Modeling the Effects of the Local Environment on a Received GNSS Signal

DISSERTATION

Marshall E. Haker, Major, USAF

AFIT/DEE/ENG/12-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

AFIT/DEE/ENG/12-11

Modeling the Effects of the Local Environment on a Received GNSS Signal

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Marshall E. Haker, B.S.E.C.E., M.S.E.E.
Major, USAF

December 2012

AFIT/DEE/ENG/12-11

Modeling the Effects of the Local Environment on a Received GNSS Signal

Marshall E. Haker, B.S.E.C.E., M.S.E.E.

Major, USAF

Approved:

/signed/

John F. Raquet, Ph.D. date
Committee Chairperson

/signed/

Matthew C. Fickus, Ph.D. date
Committee Member

/signed/

Richard K. Martin, Ph.D. date
Committee Member

/signed/

Frank van Graas, Ph.D. date
Committee Member

Accepted:

M. U. Thomas Date
Dean, Graduate School of Engineering and Management

AFIT/DEE/ENG/12-11

Abstract

GNSS receiver designers extensively use models to describe the effects of the local environment, particularly multipath, on a received GNSS signal. These models are used to construct simulators in order to generate data that can be used to test receiver signal processing techniques. Though there are many models and simulators available which feature multipath effects, the legitimacy of these models is questionable, depending on how the model is constructed. There is an ongoing need in the GNSS community for the development of high-fidelity simulators which generate data that replicates what can truly be expected from a challenging environment such as an urban canyon or an indoor environment. The algorithm developed for use in the research in this dissertation, the Signal Decomposition and Parameterization Algorithm (SDPA), is presented in order to respond to this need. This algorithm is designed to decompose a signal received using a GNSS recording and playback system and output parameters that can be used to reconstruct the effects on the signal of the environment local to the receiver at the time of recording.

There are two principal contributions to the community made with this research. First, the SDPA itself is presented and compared with what is believed to be the state-of-the-art in GNSS multipath parameterization, a Space Alternating Generalized Expectation Maximization (SAGE) algorithm. Second, the development and characterization of a stopping criteria that can be used to halt the SDPA when parameterization of salient components within a recorded signal has been completed.

Presented in this document is the finding that, for an integration period equal to the duration of a complete GPS C/A code sequence (1 msec), SDPA outperforms a SAGE-based algorithm at a carrier-to-noise ratio C/N_0 of 40 dB-Hz or below. This finding exposes the existence of scenarios where decomposition of GNSS received signals is more appropriately performed using SDPA rather than with SAGE.

Acknowledgements

I'm grateful. I'm grateful to God. I'm grateful for the truth of **Romans 8**.

I'm grateful to my research advisor, Dr. John Raquet, for choosing me. But more importantly, for being a very decent, very thoughtful, very thorough, very kind human being who worked relentlessly to coach me up on what's required of a doctoral student, faculty member, and researcher. *The ride was worth it. Thanks, boss.*

I'm grateful to my committee members, Dr. Rick Martin, Dr. Matt Fickus, and Dr. Frank van Graas, all of whom contributed to this document and to the process by providing constructive feedback without ever trying to discourage me.

I'm grateful to the people of the Advanced Navigation Technology Center, previous and present: Dr. Mike Temple (who bailed me out), Lt Col Mike Veth, Lt Col Ken Fisher, Lt Col Mike Stepaniak, Mark Smearcheck, Tom "Swag" Pestak, Jared Kresge, Fred Webber, Tina Cole, Nicole Elkins, Kaleena Glover, and Ben Leach.

I'm grateful to Lt Col Paul Harmer and Run Club, Lt Col Kyle Kolsti, Major Jeremiah Shockley, Major Jose Gutierrez, Capt Eric Buschelman and Mr. Jim Brewer. These guys supported me and kept me level-headed when stuff wasn't 100% fun.

I'm grateful to a handful of faculty members associated with the navigation community within several Ohio universities, most notably Dr. Sanjeev Gunawardena, Dr. Wouter Pelgrum, Dr. Maarten Uijt de Haag, and Dr. Jade Morton.

I'm grateful for the people of both campus320 and Crestview. Prayer matters.

I'm grateful for my parents, my brothers, and my in-laws. Thank you.

I'm grateful for my delightful children. They both radiate with purpose.

I'm grateful for my wife. She is the warmest, kindest, most thoughtful, most deliberately constructive and best influence on all that has become of my life. Love, *kingdom lights shine.* ***I had the time of my life fighting dragons with you.***

Marshall E. Haker

Table of Contents

Page

List of Figures

Figure Page

Figure Page

List of Tables

Modeling the Effects of the Local Environment on a Received GNSS Signal

I. Introduction

This dissertation presents the Signal Decomposition and Parameterization Algorithm (SDPA), as well as the stopping criteria required in order to halt processing of SDPA. This iterative algorithm is designed to decompose recorded Global Navigation Satellite System (GNSS) signals in order to obtain estimates associated with both direct path and multipath ray waveforms. As part of this research, a stopping criteria is developed for use in halting decomposition iterations.

This chapter provides a purpose statement in Section 1.1, a brief statement on the background behind this research in Section 1.2, a summary of the specific problem at hand addressed with this research in Section 1.3, the scope of the research in Section 1.4, and an overview of the dissertation in Section 1.5.

1.1 Purpose

This document presents two principal contributions to the body of work in the area of GNSS receiver signal processing, modeling, and simulation. The first contribution is the development of SDPA. SDPA is a signal decomposition algorithm that offers, in limited and specific scenarios, the ability to better reduce the RMS error between the true signal and the estimated signal when transformed to a time-frequency domain (using the short-time Fourier transform) versus the state of the art. The second contribution is the development of a stopping criteria to be used in conjunction with SDPA.

1.2 Background

A GNSS, such as the Global Positioning System (GPS), uses signals transmitted from satellites to obtain precise timing information at a receiving antenna. These signals are subject to many error sources, such as ionospheric and tropospheric effects, that distort the originally transmitted signal. In position computations conducted in modern GNSS receivers, a major source of error is typically attributed to the impact that features in the local environment have on received signals, to include multipath propagation. The features in the environment local to the GNSS receiver antenna distort the line-of-sight (LOS) signal required to accurately obtain the pseudorange measurements used in computing the position of the receive antenna. Different environments will have different effects on received GNSS signals. The dominant error source associated with these local environmental effects is referred to as multipath propagation, the reception of replicas of the transmitted signal that have traversed multiple, varying paths between the transmitter and receiver. The timing of GNSS signal reception is critical to accurate positioning, and the presence of multipath in a received signal degrades the accuracy of signal reception timing estimation. Therefore, multipath continues to be addressed in GNSS research.

GNSS receiver signal processing technique designers use multipath models extensively as they work to reduce geopositioning errors. These models are used to construct simulators in order to generate data that can be used to test these receiver signal processing techniques. Though there are many models and simulators available which feature multipath effects, the legitimacy of these models is sometimes questionable, depending on how the model is constructed. There is an ongoing need in the GNSS community for the development of high-fidelity simulators which generate data that replicates what can truly be expected from a challenging environment such as an urban canyon or an indoor environment. The algorithm developed for use in the research in this dissertation, SDPA, is presented in order to respond to this need. This algorithm is designed to decompose a signal received using a GNSS recording and playback system and output parameters that can be used to reconstruct the effects on the signal of the environment local to the receiver at the time of recording.

1.3 Problem Statement

As of now, the fidelity made available with GNSS simulators is limited. The problem that arises because of this is that techniques developed to mitigate the effects of the local environment on received GNSS signals cannot be effectively tested. This is because these techniques are tested using simulated signals, rather than signals impacted by real-world environmental effects. Because of this discrepancy between simulated and real-world signal characteristics, mitigation techniques are often developed and verified in the lab, only to be deployed for use unsuccessfully because of the model's lack of adherence to the characteristics of the true environment. This problem may be alleviated through the use of a recording and playback system. A system such as this is used to record real-world GNSS signals, and then replay these recorded signals. This is distinct from a simulator that uses models to construct signals that demonstrate propagation effects that would be present in the channel. The advantage of a recording and playback system is that there is no uncertainty regarding the validity of the effects observed in the recorded signal. The disadvantage of this system is that no models are obtained that can be used in mathematically describing the channel. The goal in this research is to bridge the gap between signals generated using mathematical models and recorded real-world signals, through the use of a hybrid system. This hybrid system offers both the advantages of a recording and playback system and a system that uses an analytic model to generate data. Propagation effects contained in recorded data would be analyzed in software in order to obtain statistics for describing these effects. These statistics could then be used to model propagation effects.

A condition placed on this hybrid system is that each segment of received data be considered independently from other segments of received data in the estimation of parameters. The independent consideration of each data segment eliminates the possibility that errors in the estimation of parameters from previously considered data segments will impact the estimation of parameters from the data segment currently being considered, as may be the case when using an architecture that employs tracking loops.

This hybrid system makes use of an iterative signal decomposition methodology. Since the approach is iterative, there is a need to develop a stopping criteria to direct the algorithm to halt decomposition. If there is no stopping criteria developed, time is wasted on processing the received signal beyond the point where decomposition outputs are optimal, or even meaningful.

1.4 Scope

In this research, only GPS civil signals (L1 C/A-coded signals) will be considered, though the results obtained in this work will generally apply to other GNSS signal types, or generalized wireless communications signaling that makes use of direct sequence spread spectrum waveforms. GNSS signal parameter estimation error sources that are considered in this dissertation include only additive white Gaussian noise and GNSS signal replicas to serve as multipath waveforms, as would be found in conventional GNSS signal plus multipath models.

1.5 Dissertation Overview

A survey of literature on the modeling and simulation of GNSS signals, on multipath propagation, and the state-of-the-art in multipath parameter estimation for GNSS (application of the Space-Alternating Generalized Expectation Maximization algorithm, or SAGE) is provided in Chapter II. The Signal Decomposition and Parameterization Algorithm is described in canonical detail and compared with SAGE, in terms of methodology, in Chapter III. Results of comparisons between SDPA and SAGE are provided in Chapter IV with an emphasis on the scenario in which SDPA outperforms SAGE. Chapter V outlines the methodology and the characterization of the stopping criteria designed for use in SDPA, with additional consideration of application of the same stopping criteria in conjunction with a SAGE-based decomposition engine. This dissertation is concluded in Chapter VI, and includes recommendations for future research.

II. Background

2.1 Introduction

This chapter provides the background theory that underpins the research findings reported in this dissertation. A general analytic GNSS signal model, both transmitted and received (with and without multipath propagation effects), is provided in Section 2.2. Section 2.3 briefly describes how GNSS signals are processed in standard receivers, in terms of both the analog-to-digital pre-processing of the received signal and the processing to obtain the measurements required to compute the position of the receive antenna. Section 2.4 provides background from literature in the areas of multipath propagation and the effect that multipath has on GNSS signaling. Section 2.5 provides background information for the techniques used in processing GNSS signals in this research, which are generally used to parameterize multipath signatures contained in the received GNSS signal and to track the variations in these signatures over time. Section 2.6 outlines the theory behind and framework for the GNSS receiver signal processing technique called the Space Alternating Generalized Expectation Maximization (SAGE) algorithm, which is the existing algorithm that most closely resembles the SDPA technique developed for the research outlined in this dissertation. Section 2.7 provides background on the Multipath Estimating Delay Locked Loop receiver design, an approach to multipath parameter estimation that is similar to SAGE but requires use of a delay locked loop structure. Section 2.8 provides the theoretical foundation used in the development of the stopping criteria developed as part of SDPA. Background information on the current state of GNSS simulation is provided in Section 2.9. Finally, background information on the structure of GNSS signals from global constellations other than GPS is presented in Section 2.10.

2.2 Analytic Models of GNSS Signaling

In this research, only GPS coarse acquisition (C/A-coded) civil signals are considered (versus other radionavigation signals, such as GPS military (P(Y)), Compass, or GLONASS signals), so discussion of analytic expressions will be limited to the

C/A-coded waveform. A model is first presented for the transmitted signal in Section 2.2.1, then a model for the GPS signal received in a noisy environment with no multipath effects in Section 2.2.2, and then a model for the GPS signal received in a noisy environment with multipath effects in Section 2.2.3.

To describe GPS signal models analytically, the following equations and notation are used. To achieve the accuracy GPS potentially offers, one must be able to process GPS signals that are transmitted in the continuous-time domain through the wireless channel. These signals therefore are expressed in notation making use of a continuous, as opposed to a discrete, time variable, t. However, all of the signal processing in the software-defined receivers employed in this research deals with time exclusively in the discrete (sampled) domain, where a variable such as n is used. It therefore becomes necessary to map variables and equations from the continuous to the discrete domain in order to ensure that analytic expressions from multiple domains are unified in functional outcome. The analytic models presented in [1] will additionally have to be adapted to accommodate multipath propagation. The adaptation for the presence of multipath-propagated waveforms will make use of the theory found in digital communications literature [2] [3]. Before this is done, the fundamental GPS signaling equations will be presented.

GPS radionavigation, which is a subset of GNSS radionavigation, is defined for the purposes of this prospectus as the communication of signals that are transmitted from the GPS space segment, the GPS space vehicles (SV) that orbit the earth specifically to transmit radionavigation signals. These radionavigation signals are received by users composing the GPS user segment after communication through the wireless channel between the satellite and the earth. Terrestrial users are considered for the results outlined in this dissertation, more specifically ground-based fixed or mobile receivers (either pedestrian or automobile mounted). This does not, however, preclude the application of this research to receivers onboard aircraft.

The operational GPS constellation originally made primary use of two center frequencies, referred to as L1 and L2, to transmit radionavigation signals. These two frequencies are $f_{L1} = 1575.42$ MHz and $f_{L2} = 1227.60$ MHz [1] respectively. The frequency of interest for this research is f_{L1} and is referred to as L1 for the remainder of this document.

The GPS radionavigation signal transmitted on L1 consists of a signal that is open and can be decoded for use by civil users, and a signal that is protected for use only by DoD-authorized users. The open signal provides what is called the Standard Positioning Service (SPS) [4], and the protected signal provides what is called the Precise Positioning Service (PPS) [5]. The key feature distinguishing the PPS from the SPS for the signals transmitted on L1 is the specific pseudorandom noise (PRN) sequences modulated within these waveforms. The SPS PRN sequence itself is available to the public for use in civil satellite navigation applications, such as GPS for transportation, whereas the PPS PRN sequence is protected. Furthermore, the SPS PRN sequence if of length 1023 chips, and repeats every millisecond [1]. The PPS PRN sequence has a chipping rate of 10.23 MHz and repeats every week. Only the L1 SPS signal is considered for this research, so only the SPS PRN sequence is described. The PRN sequence used by the SPS is referred to as the coarse acquisition (C/A)-coded sequence. Each GPS SV transmits a different C/A code (numbered simply from one to 32, so different PRN sequences are referred to as PRN 1, PRN 2, etc.), which enables the terrestrial user of the GPS radionavigation signals to distinguish between signals sent from different SVs. Each C/A-coded PRN sequence consists of a sequence of $N_c = 1023$ bits [4], referred to as chips in spread spectrum communication literature, which is used to provide differentiation between radionavigation signals from different SVs, and to provide relatively accurate timing information to GPS users. The C/A-coded PRN sequence is repeated every millisecond.

Each GPS C/A-coded signal transmitted on L1 also contains a navigation data message. The message provides information to GPS users on the position of the transmitting satellite, as well as other salient parameters. The message is transmitted using a binary phase shift-keyed digital modulation (which is discussed generally in [2] and [3], and will not be further explained here), so the navigation data message will simply consist of a bitstream that is translated into digital communications symbol values. For each bit in the navigation data message, there is a distinct digital communications symbol associated with the bit value. There are two possible bit values (0 and 1), therefore there are two possible symbol values (-1 and +1). These symbol values are transmitted at a rate of 50 bits per second [1]. The leading edge of a navigation data symbol is aligned with the leading edge of the start of the first of the $N_c = 1023$ chips in a C/A-coded PRN sequence. The trailing edge of a navigation data symbol is aligned with the trailing edge of the last of the 1023 chips in a C/A-coded PRN sequence. Therefore, exactly 20 instances of the C/A-coded PRN sequence can be found in each of the navigation data message symbols.

2.2.1 Transmitted Signal. The equation for the transmitted GPS L1 C/A-coded signal from the kth of 32 operationally-signaling satellites, $s_{L1}^{(k)}(t)$, are as follows [1]:

$$s_{L1}^{(k)}(t) = \sqrt{2P_C}x^{(k)}(t)D^{(k)}(t)\cos\left(2\pi f_{L1}t + \phi_t^{(k)}\right) \tag{2.1}$$

where P_C is the average power of the GPS L1 C/A-coded signal, $\sqrt{2P_C}$ is the peak amplitude of the transmitted GPS L1 C/A-coded civil signal, $x^{(k)}(t)$ is the value of the chip within the C/A-coded PRN sequence used for spread spectrum modulation for the kth PRN at time t, $D^{(k)}(t)$ is the navigation data symbol value for the kth PRN at time t, f_{L1} is the center frequency of the L1 signal transmitted by a GPS satellite (1575.42 MHz), and $\phi_t^{(k)}$ is the carrier phase offset of the L1 signal for the kth PRN.

Through the remainder of this description, the equations will be treated as having originated from a single SV chosen in advance, so notation for the kth SV is neglected furthermore in this document. Therefore, the above equation is restated as follows:

$$s_{L1}(t) = \sqrt{2P_C}x(t)D(t)\cos\left(2\pi f_{L1}t + \phi_t\right), t \geq 0, t \in \mathbb{R} \tag{2.2}$$

Note the constraint that $t \geq 0$. This constraint is not necessary to describe GPS signals, and is generally not found in literature on the subject, but is used later in this development (2.14)

As stated above, the C/A-coded transmission is modulated using the C/A code signal $x(t)$. The C/A waveform $x(t)$ consists of a series of pulses, referred to as chips, that are varied at a frequency of 1023 chips per msec. Since the C/A chip sequence is of length 1023, this means that the chip sequence repeats every msec. The C/A chip sequence is generated using Gold codes [1], and is expressed as follows:

$$x(t) \in \{1, -1\} \tag{2.3}$$

One period (of duration 1 msec) of the C/A code can be expressed as follows [1]:

$$x(t) = \sum_{n=0}^{N-1} x_n p \left(\frac{t - nT_c}{T_c} \right) \tag{2.4}$$

where $N = 1023$ chips, x_n is the value of the nth element in the vector of chip values in the C/A PRN sequence of length 1023, $p(t)$ is a pulse waveform (with unit height, unit width, and centered at the origin), and $T_c = 1/1.023$ μsec and is the duration of any chip in the C/A PRN sequence.

The C/A-coded signal is also modulated using a data sequence $D(t)$. The data sequence is transmitted at a frequency of 50 bits per sec, in alignment with the C/A-coded chips (so the beginning and end of data bits align in time with the beginning and end of frames, respectively). The set of possible data sequence values is expressed as follows:

$$D(t) \in \{1, -1\} \tag{2.5}$$

The initial carrier phase ϕ_t can vary arbitrarily as follows, as a function of any phase shifts that occur within the transmitter circuitry in the SV.

$$\phi_t \in [-\pi, \pi) \tag{2.6}$$

2.2.2 Received Signal in the Absence of Multipath Effects. When the signal is transmitted through the wireless channel, several channel effects will impact the structure of the transmitted signal. These effects include the loss of power as the signal propagates away from the transmitter, referred to in literature as path loss; the addition of white Gaussian noise due to thermal noise within the receiver and natural noise from sources external to the receiver [1]; the variation in velocity of transmission of portions of the signal broadcast at different frequencies, particularly through the ionosphere, and referred to as code-carrier divergence; and the shift of the signal in frequency due to the Doppler effect among other phenomena [1]. These effects are contained intrinsically in the model of the LOS received signal $r_{LOS}(t)$ expressed as follows [1]:

$$r_{LOS}(t) = \sqrt{2P_r}D(t-\tau)x(t-\tau)\cos\left(2\pi\left(f_{L1}+f_D\right)t+\phi_r\right)+\eta(t) \tag{2.7}$$

where P_r is the average power of the received signal ($P_r << P_C$), $D(t-\tau)$ is the value of the navigation data message symbol at time $t-\tau$, τ is the estimated time of signal propagation between the transmitter and receiver (propagation delay), $x(t-\tau)$ is the value of the C/A-coded chip at time $t-\tau$, f_D is the value of the Doppler frequency offset from L1 incurred by the propagated signal, ϕ_r is the carrier phase offset of the received signal, and $\eta(t)$ is the white Gaussian noise added at time t to the signal during propagation. The global delay parameter τ in this equation is used to communicate the delay in propagation between signal reception and transmission. Estimation of this delay yields what is called the pseudorange, and provides the subsequent numerical basis for the ubiquitous ability to compute position using satellite radionavigation signals.

2.2.3 Addition of Multipath Propagation Effects to Received Signal . The model for the received signal expressed in (2.7) does not sufficiently describe the received signal in the presence of multipath. The model must therefore be extended to incorporate the effects of multipath propagation. To do this, the traditional ray propagation model is used, as found in communications literature [2] [3]. The multipath signal will be treated simply as one that has propagated through a path other than the LOS path. This second propagation path will have the potential to vary the average power of the multipath signal relative to the LOS signal, the potential to induce an additional Doppler shift upon the originally Doppler-shifted LOS signal, the potential for addition of further delay than what is incurred with the LOS signal, and the potential to shift the carrier phase relative to the LOS signal. Furthermore, there is the potential for an arbitrary number of non-LOS paths to propagate the broadcasted signal toward the receive antenna. To communicate this possibility, the following model for describing the complete received signal $r(t)$ is used [3]:

$$r(t) = r_{\mathrm{LOS}}(t) + r_M(t) + \eta(t) \tag{2.8}$$

where

$$r_M(t) = \sum_{m=1}^{M} \alpha_m \sqrt{2P} D(t-\tau_m) x(t-\tau_m) \cos\left(2\pi \left(f_{L1} + f_{D_m}\right) t + \phi_{r_m}\right) \tag{2.9}$$

In these two equations, m expresses the mth of M multipath rays, α_m is a scaling term expressing the attenuation of the mth multipath ray relative to the LOS ray, $D(t-\tau_m)$ is the value of the navigation data message symbol at time $t-\tau_m$, τ_m is the time of propagation incurred by the mth multipath ray, $x(t-\tau_m)$ is the value of the C/A-coded chip at time $t-\tau_m$, f_{D_m} is the Doppler shift incurred by the mth multipath ray, and ϕ_{r_m} is the carrier phase shift incurred by the mth multipath ray.

Alternately, the model can be expressed as follows, treating the LOS signal as the first ray in the summation:

$$r(t) = \sum_{m=0}^{M} \alpha_m \sqrt{2P} D(t-\tau_m) x(t-\tau_m) \cos\left(2\pi \left(f_{L1} + f_{D_m}\right) t + \phi_{r_m}\right) + \eta(t) \tag{2.10}$$

where the LOS ray is the 0th multipath ray, $\alpha_0 = 1$, $\tau_0 = \tau$, $f_{D_0} = f_D$, and $\phi_{r_0} = \phi$. This model will be used to fundamentally describe the received signal throughout the remainder of this document.

2.3 Simple Receiver Processing of GNSS Signals

Simple software-defined GNSS receivers make use of two separate processes. The first is the analog-to-digital pre-processing that is performed within what is typically referred to as the "RF front end" processing. This first step will be conducted every time real-world data is considered as part of this research, and is outlined in Section 2.3.1. The second is the process of obtaining measurements required to compute the position of the receive antenna. In a software-defined receiver, this second process is performed using software written to direct the execution of measurement generation. This second step is not researched for this dissertation, though elements of this step will be used to divide the GPS signal into C/A code frames, and is discussed in Section 2.3.2.

2.3.1 Analog-to-Digital Pre-Processing. There are two subprocesses undertaken by a software-defined GNSS receiver in order to have digital data that can be manipulated using software. The first subprocess is downconversion of the received RF signal to an intermediate frequency (IF), as described in Section 2.3.1.1. The second subprocess is sampling of the downconverted IF signal, as described in Section 2.3.1.2.

2.3.1.1 Downconversion of the RF Signal to an Intermediate Frequency. Upon receiving the signal that has been subjected to channel effects (multipath interference and other propagation effects), the signal is downconverted and filtered [1]. This is true in the case of the GPS receiver considered in this research, the Transform-Domain Instrumentation GNSS Receiver (TRIGR) [6]. Assuming the filter to be an ideal bandpass filter, the model for the received, downconverted, and filtered signal is expressed as follows:

$$r_{IF}(t) = \sum_{m=0}^{M} \alpha_m \sqrt{C} D(t - \tau_m) x(t - \tau_m) \cos\left(2\pi\left(f_{IF} + f_{D_m}\right) t + \phi_m\right) + \eta(t) \tag{2.11}$$

where C is the average power of the received LOS signal at the input into the analog-to-digital converter (ADC) inside the receiver, f_{IF} is the intermediate frequency of the receiver, and ϕ_m is the carrier phase of the mth multipath ray at the input into the ADC. Note at this point that the carrier phase of the multipath ray input to the ADC may be completely different from the carrier phase of the original received signal. This does not impact how the signal is processed. Again, the equation above is a model that makes use of the assumption of ideal bandpass filtering in its development.

This process of receiving, downconverting, and filtering is assumed to be true in the case of signals recorded using the TRIGR receiver, but it does not sufficiently describe the downconversion and filtering process for a receiver that performs in-phase and quadrature sampling. Therefore, the following models are assumed for the in-phase and quadrature signals, $i_{IF}(t)$ and $q_{IF}(t)$ respectively, that have been received, downconverted, and filtered.

$$i_{IF}(t) = \sum_{m=0}^{M} \alpha_m \sqrt{C} D(t - \tau_m) x(t - \tau_m) \cos\left(2\pi\left(f_{IF} + f_{D_m}\right) t + \phi_m\right) + \eta_i(t) \tag{2.12}$$

and

$$q_{IF}(t) = \sum_{m=0}^{M} \alpha_m \sqrt{C} D(t - \tau_m) x(t - \tau_m) \sin\left(2\pi\left(f_{IF} + f_{D_m}\right) t + \phi_m\right) + \eta_q(t) \tag{2.13}$$

where $r_{IF}(t)$ and $q_{IF}(t)$ both have independent noise terms $\eta_i(t)$ and $\eta_q(t)$ added to them. This is because both the in-phase signals and quadrature signals will have noise added to them.

2.3.1.2 Sampling of the Intermediate Frequency Signal. Sampling of the generalized model for a multipath-laden received and downconverted signal

$r_{IF}(t)$ requires that sampling theory first be discussed. Sampling is assumed to be performed periodically with an ideal ADC, so a model of a sequence of samples of the continuous-time signal $r_{IF}(t)$, the discrete-time expression of which would be $r[n]$, is as follows [7]:

$$r[n] = r_{\mathrm{IF}}(nT_s), n \in \mathbb{N} \tag{2.14}$$

where n is the sample index and T_s is the sampling period. Note that because of the presence of the baseband filter in-line between the antenna and the ADC, it is assumed that there is no aliasing present in $r_{IF}[n]$.

As with the downconversion and filtering processes, the sampling process above is assumed as true in the case of signals recorded using the TRIGR receiver, but it does not sufficiently describe sampling using a receiver that performs simultaneous sampling of the in-phase and quadrature signals. The sequences of samples of the continuous-time in-phase and quadrature signals, $i_{IF}(t)$ and $q_{IF}(t)$, the discrete time expressions of which would be $i[n]$ and $q[n]$ respectively, will be modeled as follows:

$$i[n] = i_{IF}(nT_s), n \in \mathbb{N} \tag{2.15}$$

and

$$q[n] = q_{\mathrm{IF}}(nT_s), n \in \mathbb{N} \tag{2.16}$$

Again, as with the model for $r[n]$, it is assumed there is no aliasing present in $i_{\mathrm{IF}}[n]$ and $q_{\mathrm{IF}}[n]$.

Equivalently, the in-phase and quadrature signals $i[n]$ and $q[n]$ can be approximately treated as having been generated by performing a Hilbert transform of the received signal $r[n]$ [2] [7]; where the real signal is simply shifted in phase $\pi/2$ radians; to obtain the imaginary signal. This use of the real and imaginary signals yields the in-phase and quadrature signals. The same information can be obtained from the received GPS signal regardless of whether only the real signal is sampled, as in the TRIGR receiver, or the real and imaginary signals are simultaneously sampled.

Wherever a process must accommodate a specific type of sampling, the process will highlight the accommodation in this document.

Furthermore, through the remainder of this document, equations are attempted to be expressed without being constrained by whether or not the sampling is being conducted with simple real-valued received signals or with in-phase and quadrature signals. To alleviate this, $r[n]$ is expressed to mean either $r[n] \in \Re\, \forall n$ or the following:

$$r[n] = i[n] + jq[n] \tag{2.17}$$

where $j = \sqrt{-1}$.

The general analytic description for receiver signal processing found in [1] expresses the receiver design in terms of continuous time. This conveys the receiver process, but is not sufficient for purposes of this description. Because the received signal is sampled, as well as all software processes in the research reported in this document, are executed in discrete time, future equations will be expressed in a framework using discrete time notation.

Given the real-valued or complex-valued signal $r[n]$ (so $r[n] \in \{\mathbb{R}, \mathbb{C}\}$, again depending on how the received signal is sampled), $r[n]$ can now be expressed as follows for $r[n] \in \mathbb{R}$:

$$\begin{aligned} r[n] \;=\; & \sum_{m=0}^{M[n]} \alpha_m \sqrt{C} D(nT_s - \tau_m) x(nT_s - \tau_m) \cos\left(2\pi \left(f_{\mathrm{IF}} + f_{D_m}\right) nT_s + \phi_m\right) \\ & + \eta(nT_s) \end{aligned} \tag{2.18}$$

where $M[n]$ is the number of multipath rays (which does not including the LOS ray) present at discrete-time index n. By doing this, the implicit assumption in the continuous-time equations that there is a fixed number of multipath rays for all values of t is relieved. The above equation for $r[n] \in \mathbb{C}$ can now be expressed as follows:

$$\begin{aligned} r[n] \;=\; & \sum_{m=0}^{M[n]} \alpha_m \sqrt{C} D(nT_s - \tau_m) x(nT_s - \tau_m) \cos\left(2\pi \left(f_{\mathrm{IF}} + f_{D_m}\right) nT_s + \phi_m\right) \\ & + j \sum_{m=0}^{M[n]} \alpha_m \sqrt{C} D(nT_s - \tau_m) x(nT_s - \tau_m) \sin\left(2\pi \left(f_{\mathrm{IF}} + f_{D_m}\right) nT_s + \phi_m\right) \\ & + \eta_i(nT_s) + j\eta_q(nT_s) \end{aligned} \tag{2.19}$$

The equation for $r[n] \in \mathbb{C}$ can be further simplified as follows, using Euler's relationship [8]:

$$\begin{aligned} r[n] \;=\; & \sum_{m=0}^{M[n]} \alpha_m \sqrt{C} D(nT_s - \tau_m) x(nT_s - \tau_m) \exp\left(j\left(2\pi \left(f_{\mathrm{IF}} + f_{D_m}\right) nT_s + \phi_m\right)\right) \\ & + \eta_i(nT_s) + j\eta_q(nT_s) \end{aligned} \tag{2.20}$$

where $\exp(x)$ denotes the exponential function e^x. This equation serves as the basic modeling equation upon which multipath parameterization efforts are based. It should be noted that this development does not account for errors in downconversion to an intermediate frequency or for errors in the timing of sampling (such as an erroneous offset in frequency being generated, for jitter, or similar effects). When applying these models for instrumentation receivers, the assumption that these errors are not present in received data may perhaps be made, though this is generally not the case.

2.3.2 Obtaining Measurements for Position Computation. There are two functions that are performed in what is defined here as a simple GNSS receiver—acquisition and tracking. The first function, acquisition, involves the detection and coarse parameter estimation of satellite radionavigation signals. The coarse parameter estimation is performed to obtain rough values for the Doppler frequency and the arrival time of the leading edge of the first code chip (referred to as code phase) [1]. This is done to initiate synchronization of the reference signals in the tracking loop (both in terms of code phase and carrier phase). Upon completion of acquisition, the receiver should have available for tracking a set of PRN numbers, with associated code phase and Doppler frequency for each PRN. Acquisition typically involves use of the short-time Fourier transform (STFT), referred to in GNSS literature as the search space or search grid, and is discussed in detail in section 3.3.5.

Tracking involves the operation of two feedback loops that behave symbiotically, the phase tracking loop and the code tracking loop [1]. The phase tracking loop is referred to as a phase locked loop (PLL) that provides Doppler frequency information for each PRN that is then used in the removal of the IF and Doppler frequency content from the received signal [1]. The output of this removal is a signal consisting nominally of C/A code chips, all of which are multiplied by the navigation data message bit value (either +1 or -1). This frequency "wipeoff" from the signal is required in order to conduct C/A code tracking. This process is no different from the PLL process that can be found in a typical digital communications receiver [2] [3]. The code tracking loop is used to refine initial code phase estimates and track changes in code phase over time [1]. This is done with a delay lock loop (DLL) by aligning a replica of the PRN chip sequence transmitted by the satellite with the received signal. This will yield code phase estimates which are used to eventually compute a pseudorange. Though not required to perform the simulations outlined in Chapter III, the functionality of the code tracking loop may be applied in order to divide received signals into buffered segments of duration equal to the integration period. Divisions between segments are placed at points in received data files where the last chip in the C/A-coded PRN sequence ends and the sequence begins to repeat with the first sequence. The code tracking loop provides this information, so received signals can be pre-processed using a GPS software receiver configuration in order to obtain information regarding the start and end times of PRN sequence frames in data files. This process can be found in a digital communications receiver as well [2] [3]. The difference between the DLL in GNSS receivers is that the PRN code sequence timing information is used in subsequent position computations. In both digital communications receivers and in GNSS receivers, the DLL is used to obtain bit values (for digital communications, these outputs simply constitute data, whereas this is the navigation data message in particular in a GNSS receiver).

For every PRN that is visible to the receiver, a tracking loop is employed to output code phase and carrier phase measurements. The code phase measurements from each PRN are used to compute pseudorange measurements for each respective PRN. Given that the received signal contains satellite position data and clock offset information, satellite positions and approximations of the distance between the satellite and the receive antenna can then be made available. Computations can then be performed to eventually obtain position coordinates for the receive antenna [1]. This is how simple GNSS systems provide position information to users.

2.4 Multipath Propagation and Its Effects on GNSS Signals

GNSS systems, such as GPS, Galileo, GLONASS, and Compass, make use of an RF channel to communicate information from the transmitting satellite to the receiving user. This information comes in the form of the navigation data message, as well as the relationship in timing of the RF carrier and PRN code signals onboard the GPS transmission. Multipath is typically a major source of error in modern GNSS systems [9]. This dissertation presents research into the modeling of the effects of the local environment surrounding a receive antenna on GPS L1 C/A-coded signals. Among these effects are the addition of white Gaussian noise, the loss of signal power due to propagation of the GPS signal through a medium, and multipath propagation. There is a great deal of literature available that outlines multipath propagation research. Research areas include addressing multipath from the perspective of electromagnetic propagation, from the perspective of general wireless communications theory, and from the perspective of radionavigation signals (specifically GNSS signals). This section will touch on these perspectives, but will emphasize the theory behind multipath propagation from the perspective of GNSS signals. This section is presented to provide the reader with an understanding of the theory behind multipath propagation phenomenology.

There is a large body of research and information regarding the physics behind multipath phenomenology, to include work specifically addressing the electromagnetic (EM) phenomenology of GPS signals in a multipath environment. An example of this can be found in [10], where electromagnetic wave propagation models are used to construct a physics-based, rather than statistics-based, model of GPS multipath. While this approach provides powerful insight into what might be expected in some GNSS multipath scenarios, the modeling of multipath propagation using physics-based findings is not in the scope of this research. Other literature where a physics-based, rather than probabilistic, model is used in the context of multipath can be found [11] - [13].

To provide background in multipath phenomenology, the issue will be addressed from two perspectives. The first is from the perspective of generalized wireless communications theory in Section 2.4.1. The second is from the perspective of GNSS signals in Section 2.4.2.

2.4.1 Multipath in General Wireless Communications Theory. In Section 2.2.3, an analytic model for received GPS signals in the presence of a multipath environment is presented. This model makes use of the general linear channel models for received signals bearing the impact from multipath interference found in wireless communications literature [2] [3]. At first, this may seem inappropriate, as the impact of multipath on a cellular telephone transmission may be different from the impact on a received GPS signal transmission. However, the physical phenomenology is the same, so the propagation models and equations used to describe the effects of multipath on a wireless communications signal will describe multipath in a satellite radionavigation channel. The big difference is that, in radionavigation, the timing of arrival of waveforms at the receive antenna is of critical consequence, whereas in wireless communication, the recovery of data bit values is of critical consequence.

Given the commonality of propagation effects on wireless communications versus radionavigation signals, the general wireless communications channel model will first be considered before addressing literature on multipath in GNSS. When an RF signal is broadcast, the signal is exposed to propagation effects and interference from other transmitters [14]. There are also multiplicative noise sources that are generally present, from reflections, absorption, scattering, and diffraction. These multiplicative sources are generally subdivided into three categories: path loss, slow fading (shadowing), and fast fading (multipath) [14]. Path loss is the attenuation of a signal through a channel due to the dispersion of RF energy.

The wireless channel is presented as a time-varying linear filter described by a time-varying impulse response $h(\tau; t)$, where $h(\tau; t)$ is the response of the channel at time t due to an impulse applied at time $t - \tau$ and where τ denotes the delay [2]. Signals transmitted via an aperture that are then received via a second geographically separated aperture will most likely arrive at the receiver aperture after traveling several different paths. Signals arriving after traveling through these different paths are referred to as multipath signals. For each of these multipath signals, there is a separate carrier-phase offset that can add or subtract (constructively or destructively, respectively) from the direct path, upon arriving at the receive antenna. This is called fading [2]. The time-varying model can be used to describe a channel impacted by multipath. In wireless communications theory, the time-varying communications channel exhibiting multipath is generally described stochastically [2].

When the distance between the transmit and receive antenna is changing over time, a Doppler frequency shift occurs, and the rate of change in distance between antennas corresponds to the size of frequency offset [2]. Since GNSS satellites are always moving relative to even a terrestrial user fixed in position relative to the earth, the Doppler shift must be continuously considered. This is the case for both the direct path signal and multipath signals. This adds to the number of terms that must be considered when examining multipath signals. Multipath is now expressed using a time-varying delay term, a time-varying amplitude term, and a time-varying frequency term [2].

There are two characteristics of a radio channel associated with small-scale fading that influence the structure of received signals [3]. Multipath signals arrive at the receive antenna after traveling through varying paths, with different associated delays in arrival at the receive antenna (called a time dispersive channel), but the channel medium itself changes over time, so the impulse response of the channel itself is time varying [2]. This time variance in the channel itself is due to several factors, including (but not limited to) changes in precipitation, humidity, temperature, and solar radiation. These changes are nearly impossible to predict, leading us to characterize these channels using stochastic processes rather than deterministic models. While the model expressed in (2.20) is deterministic in nature, and is of a closed form, it will be seen in Chapter III that the means by which parameters for (2.20) are obtained is stochastic in nature.

Several concepts associated with multipath in general wireless communications theory are now presented. There are different classifications of multipath propagation tailored towards GNSS signal simulation presented in [15]. However, the following sections present the classifications and definitions for multipath propagation that will be used in this dissertation, as adapted from [14].

2.4.1.1 Tapped delay line channel model. One way to model a time-variant channel is through the tapped delay line channel model [2]. This model is made up of uniformly spaced taps, such as taps in a discrete-time filter. The spacing between taps is equal to the inverse of the signal bandwidth W. Channel tap coefficients $c_n(t)$ are mutually uncorrelated, and expressed as follows [2]:

$$c_n(t) \equiv \alpha_n(t)e^{j\phi_n(t)} \tag{2.21}$$

The length of the delay line is called the multipath spread T_m, and corresponds to the amount of time dispersion in the channel [2], and is equal to L/W, where L is the maximum number of possible multipath signals. A mathematical expression describing the received signal $r(t)$ output from the time-varying tapped delay line channel model is as follows [2]:

$$r(t) = \sum_{l=1}^{L(t)} \left[s(t) \frac{c_l(t)}{W^{l-1}} \right] + \eta(t) \tag{2.22}$$

where l is the tap number, $L(t)$ is the total number of taps at time t, $s(t)$ is the transmitted signal at time t, $c_l(t)$ is the value of tap l at time t, and $\eta(t)$ is the value of the additive white Gaussian noise at time t. The tapped delay line model is used in digital communications theory to describe channels.

2.4.1.2 Fading. Fading is described as signals arriving at a received antenna after traveling several different paths [14]. Fading will yield increases and decreases in the power level of the received signal due to constructive and destructive interference, respectively. Fast fading is on the scale of half of a wavelength [14]. As an obstruction increases in size, so too does the effect of the obstruction. Tap coefficients in the tapped-delay line model can be modeled as complex-valued Gaussian random processes. Tap coefficients $c(t)$ can be expressed as follows [2]:

$$c_l(t) = c_{l_r}(t) + jc_{l_i}(t) = \alpha_l(t)e^{j\phi_l(t)} \tag{2.23}$$

where $c_{l_r}(t)$ and $c_{l_i}(t)$ are real-valued Gaussian random processes. These processes are generally assumed to be stationary and statistically independent. If $c_r(t)$ and $c_i(t)$ are zero-mean Gaussian signals, the amplitude $\alpha(t)$ is described using the Rayleigh probability distribution [2]. $\phi(t)$ is then uniformly distributed over $[0, 2\pi)$. This yields the Rayleigh fading channel, which is modeled using the following probability distribution function (PDF) [2]:

$$f(\alpha) = \begin{cases} \frac{\alpha}{\sigma^2} e^{-\alpha^2/2\sigma^2}, & \alpha \geq 0 \\ 0, & \alpha < 0 \end{cases} \tag{2.24}$$

f(n) =

where $\sigma^2 = E(c_r^2) = E(c_i^2)$. Rayleigh fading, a type of small-scale fading, takes place if there is no dominant LOS, or direct, path between the transmitter and receiver antennas (put simply, the LOS signal path is blocked) [3]:

If one or both of the processes $c_r(t)$ and $c_i(t)$ are not of zero mean, then $\alpha(t)$ is modeled using the Rician PDF [2]. The phase $\phi(t)$ will have a nonzero mean as well. This fading channel is then called Rician. The PDF of α is as follows [2]:

$$f(\alpha) = \frac{\alpha}{\sigma^2} e^{-(\alpha^2+s^2)/2\sigma^2} I_0\left(\frac{s\alpha}{\sigma^2}\right), \alpha \geq 0 \tag{2.25}$$

where s^2 is the power of the received signal components that do not fade, and $\sigma^2 =$ VAR$(c_r) =$ VAR(c_i).

2.4.1.3 Shadowing. Signals are impacted by shadowing when there are large obstructions between the transmitter and receiver [2]. Shadowing is a large-scale form of fading, where the direct path signal is attenuated [3], yielding a larger path loss than that observed in open space signals. Terrain contours (vegetation, buildings, etc.) between the transmitter and receiver yield shadowing. Shadowing varies significantly over distances of hundreds of meters [14]. The received signal power varies with the nature of obstructions between the transmit and receive antenna. Shadowing decreases as the elevation angle of the satellite relative to the horizon increases [16]. Shadowing is modeled as a multiplicative, slowly time-varying random process r(t) as follows [2]:

$$r(t) = A_0 g(t) s(t) \tag{2.26}$$

where A_0 is the mean path loss, $s(t)$ is the transmitted signal, and $g(t)$ is a random process used to represent the shadowing effect. The shadowing process can be modeled as a log-normal distribution (i.e., a distribution in which the signal expressed in decibels has a normal distribution), as seen in the following equation [2]:

$$f(g) = \begin{cases} \frac{1}{\sqrt{(2\pi\sigma^2 g)}} e^{-(\ln(g)-\mu)^2/2\sigma^2}, & g \geq 0 \\ 0, & g < 0 \end{cases} \tag{2.27}$$

A mean path loss as a function of distance is estimated, and then the path loss is varied with a log-normal distribution about the mean [3].

2.4.1.4 Reflection, Diffraction, and Scattering. Small-scale and large-scale fading descriptions making use of Rician, Rayleigh, or log-normal PDFs are means by which the effects of propagation can be modeled. However, no insight into what in particular creates the multipath scenario is offered with these models alone. According to Sklar, there are three basic categories of phenomena that affect signal propagation in mobile communications: reflection, diffraction, and scattering [3].

Reflection occurs when a wave impinges upon a smooth surface [3]. This smooth surface must be larger in dimension than the wavelength of the RF signal. This is referred to as specular multipath [17]. If a large, smooth reflector is indeed present in a position adjacent to the direct line of sight of the direct ray, the resulting specular signal can be modeled with the following equation [17]:

$$r(t) = As(t-\tau)e^{j\phi} + \alpha As(t-\tau-\tau_m)e^{j\phi_m} + n(t) \tag{2.28}$$

where $s(t)$ is the transmitted signal, A is the amplitude of the received direct path signal, α is the relative received multipath signal amplitude, τ_m is the delay of the reflected signal relative to the direct signal, and $n(t)$ is the noise term. However, true specular multipath rarely occurs [17]. Surfaces should only be considered smooth enough to induce specular reflections if the surface phase differences are less than $\pi/8$ radians [14].

Shadowing is caused by diffraction [3]. Diffraction occurs when the signal path between the transmitter and receiver is at least partially obstructed by a body of different density than the open air environment. This body must have dimensions that are large relative to the signal wavelength. Secondary waves are formed behind the obstruction. This phenomenon makes it possible for RF radiation to travel from the transmitter to the receiver without a direct line of sight between the two.

Scattering occurs when RF radiation impacts rough surfaces or surfaces whose dimensions are small relative to the wavelength of the signal [3]. The edges of these surfaces, called scatterers, cause energy to be dispersed in essentially random directions. Typical urban obstructions that cause scattering include signage, lights, and vegetation.

2.4.2 Multipath in GNSS Signals. Multipath distorts the modulation and carrier phase of transmitted GNSS signals, which leads to a degradation in positioning accuracy [9]. Measurement errors resulting from multipath depend primarily upon the strength of the reflected signal relative to the direct signal and the delay between the direct and reflected signals, and affect both code- and carrier-phase measurements taken in a GNSS receiver [1]. Typical pseudorange measurement errors vary from one to five meters depending on the number of scatterers and reflectors in the environment [1]. Literature provides in-depth information on pseudorange measurement performance in the presence of multipath [18]. Typically, carrier-phase multipath errors are two orders of magnitude smaller than code-phase multipath errors.

As the elevation angle of a transmitting satellite gets smaller, the received signal power decreases while the multipath interference power increases [1]. Because of this, measurements from satellites that are low in elevation are often neglected altogether. The case exists when a multipath signal can arrive at approximately the same time and at the opposite phase from the LOS signal. This has the effect of reducing the signal to noise ratio [19].

To provide context for the discussion of previous research into GNSS multipath phenomenology that can be found in literature, it is important to understand the problem that is considered in this research. The motivation for this work is to progress towards the ability to generate models with which simulators can be constructed for the generation of sample GNSS data. This data should resemble received GNSS data as closely as possible, while still retaining the ability to express parametrically the models used to generate this data. In short, this research is intended to progress toward development of a hybrid system that offers both the advantages of a recording and playback system and a system that uses an analytic model to generate data. There are several GNSS recording and playback systems that are available commercially today [20]. However, a system that simply records a GNSS signal and plays the signal back does not additionally offer estimates for various propagation effects in the channel that could be used in modeling and mitigation. To construct a hybrid system making use of the findings presented in this dissertation, it is important to have an understanding of the previous work that has be conducted in the areas of GNSS multipath estimation and GNSS multipath modeling. There is a difference between the two that may be worth clarifying. Multipath estimation is the processing of real-world data to obtain statistics for describing the multipath in the environment. Multipath modeling is analytically describing what is expected to be observed in when recording GNSS signals, using equations and parameters founded in electromagnetic propagation and wireless communication theory. A recording and playback system offer neither multipath estimation nor multipath models. A simulator that makes use of multipath models may make use of some limited set of multipath estimates that have been obtained previously to the development of the model. The research reported in this dissertation considers a scenario where GPS data is recorded, signal processing techniques are applied to obtain estimates for various parameters used in describing multipath phenomenology, models making use of these estimates are constructed, and a simulator that makes use of these models in order to generate an infinite amount of data is finally constructed. This process of recording, estimating, modeling, and simulation of GNSS multipath in a comprehensive system has yet to be found in GNSS literature on the subject.

The scope of the background content for GPS multipath will first be presented as follows:

- GPS multipath in particular will be considered. There is literature in the area of multi-constellation GNSS radionavigation in the presence of multipath [21] [22], but multi-constellation radionavigation will not be considered in the research presented in this dissertation. There is also literature that specifically addresses the signals associated with constellations other than GPS [23] [24]. There are similarities in phenomenology between GPS signals and signals from other realized or notional global navigation constellations (such as Galileo), but there are also differences. Galileo in particular uses a binary offset carrier modulation within its civil service signal structure. This requires specific considerations be made to accommodate the estimation of the multipath channel [25].
- Multipath will only be considered in the context of signals that are received and processed using a single receiver with a single antenna consisting of a single aperture, that is generally assumed for discussions in this document to be isotropic. There is a distinct body of research separate from the research in this dissertation that involves the study of multipath-laden signals obtained by receivers making use of multiple apertures that permit techniques like beamforming to take place [26] - [31].
- The estimation and modeling is considered at the pre-detected (intermediate frequency sampled signal) level, rather than at the level involving carrier phase or pseudorange measurements. There is literature available that considers multipath on pseudorange and carrier phase measurements, rather than at the level of pre-detected signal samples [32] - [40].
- Only one GPS RF carrier frequency, L1, is considered in this research in the estimation, modeling and simulation of GPS signals laden with multipath. This

does not, however, preclude the findings from being applied to other frequencies or to multi-carrier GNSS signal processing techniques. Relative to the research outlined in this dissertation, there is a dissimilar body of literature discussing the simultaneous use of multiple carriers in the consideration of GPS signals in the presence of multipath propagation [41].

Literature states that there are two broad categories of techniques by which multipath is mitigated within GNSS receivers—techniques that do require that multipath be estimated and techniques that do not require multipath estimation [42] [43]. The algorithm used in this research is designed specifically to estimate parameters associated with multipath propagated signals received with in addition to a line of sight GPS signal and noise.

There are several examples of multipath mitigation techniques that do not involves the estimation of multipath [44] [45], such as the Narrow Correlator [46], the Strobe Correlator [47], the Vision Correlator [48], and the use of a whitening filter to remove correlated noise [49]. These examples all involves techniques that work to mitigate multipath without any sort of effort to determine if multipath is present. Other techniques that function similarly to these, but additionally acts to detect the presence of multipath (without estimating multipath ray parameters) have also been developed [50] [51]. A technique that works to detect multipath through the use of signal to noise ratio data obtained from the received signal has also been developed [52]. A further extension of this approach is a category of techniques that estimate error due to multipath in order to remove the impact of the error on measurements, without actually estimating multipath rays parameters (e.g., [53]).

Techniques that estimate multipath explicitly in the act of mitigating multipath are the Multipath Estimating Delay Lock Loop (MEDLL) [54] [55], Multipath Mitigation Technology (MMT) [56], and application of the SAGE algorithm [57]. Each of these algorithms go so far as to explicitly estimate parameter values for received GNSS signal rays. Parameters that are assigned values are the ray amplitude, the ray propagation delay relative to the line of sight ray, and the carrier phase for each ray. Another technique where the amplitude and propagation delay are estimated is the Reference Correlation Multipath Mitigation (RCMPM) [58]. The use of particle filters has also been employed in the detection of the presence and estimation of parameters for multipath [59] - [62]. Frequency-domain techniques (making use of cepstral analysis and direct inverse fast Fourier transforms to obtain a channel impulse response) [63] [64] and least squares-based techniques [65] [66] for estimating multipath parameters have been developed.

2.4.2.1 Code- and Carrier-phase Multipath Ray Model. Braasch develops closed form equations for code- and carrier-phase multipath errors resulting from a single multipath ray entering a stationary receiver [9]. These equations can obviously be expanded to include multiple multipath rays in an ensemble. A broadcast GPS signal $s_1(t)$ can be expressed as follows [9]:

$$s_1(t) = A\cos\left[\omega_0 t + P(t)\pi/2\right] \tag{2.29}$$

where A is the magnitude of the received waveform, ω_0 is the frequency of the received signal (which includes the carrier frequency and any Doppler shift), and $P(t)$ is the PRN code chip value (+1 or -1).

Braasch asserts that there are four parameters that characterize multipath: amplitude of the reflected signal relative to the LOS signal, time delay of the reflected signal relative to the direct signal, phase of the relative signal relative to the direct signal, and rate of change of the relative phase between the reflected and direct signal [9]. All four of these parameters are relative to the direct signal. In a stable multipath scenario, which is assumed in further equations, it is assumed that the relative phase rate of change is equal to zero. The relative phase of the multipath is a function of both the relative time delay and the reflection coefficient of the reflecting object.

A received signal in a stable multipath scenario composed of the LOS signal and a single multipath ray is expressed as follows [9]:

$$s_{1m}(t) = -Ap(t)\sin(\omega_0 t) - \alpha Ap(t+\delta)\sin(\omega_0 t + \theta_m) \tag{2.30}$$

where α is the amplitude of the multipath signal relative to the direct signal, δ is the time delay of the multipath signal relative to the direct signal (which is always negative in the convention of the above equation), and θ_m is the multipath phase relative to the direct signal.

Another similar model that is tailored specifically for GPS is posed in [67], where the received signal consists of the direct signal and $M-1$ multipath signals. This equation is as follows [67]:

$$x(t) = \sum_{i=1}^{M} A_i(t) D\left(t - \tau_i(t)\right) C\left(t - \tau_i(t)\right) \cos\left(2\pi(f_0 + v_i(t))t + \psi_i(t)\right) + n(t) \tag{2.31}$$

where $A_i(t)$ is the amplitude of the ith multipath component, D is the amplitude of the navigation message, C is the amplitude of the PRN sequence code value, τ is the multipath error, v_i is the frequency change, ψ_i is the phase change, and $n(t)$ is noise. With the exception of the variables used and a variation in the scaling (one variable versus two variables) of the peak amplitude of each multipath ray, the model in (2.31) is effectively the same as that expressed earlier in (2.10).

Braasch compares the closed form equations of a direct signal with the closed form equations of a direct signal summed with a multipath ray signal. These comparisons take place as both signals (the direct signal by itself, and the direct signal summed with the multipath ray signal) are processed in the forward processing (from carrier and Doppler removal to discriminator) of a coherent delay lock discriminator [9]. This same style of comparison takes place for the case of a noncoherent DLL discriminator [9]. Braasch also presents closed form equations of a combined direct and multipath ray signal as the signal is forward-processed in a carrier PLL [9].

Braasch asserts that, in cases where the absolute value of the DLL tracking error is greater than or equal to one spreading code chip, the DLL tracking loop actually tracks the multipath signal, rather than the direct signal [9].

Since carrier phase errors cannot exceed 90° when the single ray multipath signal strength is less than the direct path signal strength, the impact on carrier phase error is no more than 4.8 cm for GPS L1 [9]. This can be verified analytically with the following equation for the multipath error in the carrier phase measurement $\delta\phi$ [1]:

$$\delta\phi = \arctan\left(\frac{\sin(\Delta\phi)}{\alpha^{-1} + \cos(\Delta\phi)}\right) \tag{2.32}$$

If $\alpha < 1$, $\delta\phi = 90^\circ$ in the worst case.

When the Doppler shifts between the multipath signals arriving at different directions from the LOS signal contrast well enough from the Doppler shift exhibited in the LOS signal, the received multipath signals become decorrelated from the LOS signal path, so they can be neglected [19]. This is because these multipath signals do not correlate well with the receiver reference signal that is tracking the received direct signal.

Code phase errors can be more than 100 m, however, because of the effects of a single multipath ray when processing GPS L1 C/A-coded signals. The error envelope for the noncoherent DLL is the same as that for the coherent DLL [9]. When the indirect ray path amplitude is small compared to the direct ray path, the error due to multipath varies sinusoidally, but when the indirect ray path amplitude is large, sharp discontinuities in error may be present [9]. Pseudorange errors are at their highest when carrier-phase measurement errors are at their smallest, and vice versa [9]. The specific reason why code-phase measurement errors are pronounced in a multipath environment is because the DLL discriminator takes measurements from secondary peaks in the correlation process that distort the early and late correlator measurements. Because of this, the correlator will not center on the true arrival time of the direct ray [1] [67]. The reflected wave is asserted generally to arrive after the direct ray, and creates a delay in the correlation peak. The relative delay of the multipath relative to the signal from the direct path will act to determine the effect of the multipath signal. If the delay is large relative to the direct path signal, then the correlation process will act to suppress the power of the multipath signal and reduce its impact [1]. If the delay is greater than twice the chip time, the multipath signal is relatively easily resolved and its effect managed [19]. Severe shadowing may lead to the receiver tracking the multipath signal rather than the direct path signal. This obviously leads to large ranging errors.

2.4.2.2 Efforts in Modeling the GNSS Channel. There are a number of places where research is being conducted into GNSS modeling, estimation, and mitigation (as it relates to multipath estimation), as referenced earlier in this chapter. Research in two such places, the German Aerospace Center (DLR) and the University of Calgary, is discussed further. Modeling work performed at these locations is presented here because the efforts outlined by DLR and Calgary align most closely with the research presented in this dissertation. DLR has placed an emphasis on outdoor urban, suburban, and rural land mobile multipath, leveraging the results of a measurement campaign undertaken in the last 10 years. Calgary is studying the GNSS channel in both outdoor terrestrial and indoor settings. DLR's work on land mobile modeling and Calgary's work in indoor modeling will be discussed.

DLR has undertaken an extensive effort to model the GNSS channel in both rural and urban environments. This is to develop techniques to mitigate the local environmental effects observed in the GNSS channel, and construct a GNSS signal simulator that incorporates some of their findings into the simulator. DLR research has led to the construction of a model that characterizes the time-varying channel between a satellite and a mobile user [16]. This is done by surveying the channel and then attempting to fit parameters to the channel. Information that can be used directly in modeling the GPS channel is not necessarily provided in [16], as the L1 channel is more than 200 MHz separated from the frequency of interest in the DLR channel survey campaign. The methodology used in [16], however, does provide a framework within which results could be derived. To perform this modeling, a measurement campaign was conducted to parameterize channel characteristics which can be used in presenting a model. Measurements were performed using a carrier frequency of 1820 MHz, a transmitter onboard an aircraft simulating a satellite, and a mobile receiver. The pseudonoise bit sequence operated at a bandwidth of 30 MHz, at a sampling rate yielding 15.6 impulse responses per second. It was found that multipath echoes with small delays (less than 600 nsec) appear, and are typically attenuated by 10-30 dB. Furthermore, it was found that environments differ because of the influence of shadowing, as well as the number and attenuation of echoes. Furthermore, the power of the echoes decreases exponentially versus the delay. The number of echoes with long delays, as it turns out, is small compared with the number of echoes with short delays. Channel behavior in [16] is described through use of the following variables: the time-delay system function $h(t, \tau)$, the echo delay τ, and time t. Furthermore, the transfer function of the channel $\mathrm{CTF}(t, f) = \mathrm{FFT}_{\tau}(h(t, \tau))$ is used to describe the channel properties in the frequency domain. With CTF the frequency selectivity of the channel can be directly observed. In the ideal channel, where the environment is open, the CTF is flat over the 30 MHz system bandwidth. This is not generally realistic, however. In a more general environment, the CTF depends mainly on shadowing of the LOS signal. Echoes do not contribute very much to the signal power in LOS conditions. The suburban environment actually more closely resembles the ideal channel than it does the rural environment with trees increasing the amount of shadowing. The urban environment is more frequency selective. It is asserted that this is due to the echo reflections from traffic and buildings. It is uncovered in [16] that there is a spectral correlation of 5 Hz. This leads to the assertion that the main multipath echoes are delayed by 200 nsec (thus an indirect path length 60 m longer than the direct path). The CTF is attenuated by 10 to 20 dB in a shadowed environment, either rural or urban. The channel becomes frequency selective at a distance from the carrier frequency of 3 to 5 MHz.

In research conducted by DLR, a channel model was developed that is founded on an interpretation of the physical environment. This model assumes several components [16]:

- Different reflectors R_k yield echoes with a round-trip detour Δs_k and a delay $\Delta\tau_k = \Delta s_k/c_0$. These parameters are with respect to the propagation delay of an undisturbed signal.
- The reflected components of the transmitted signal undergo superposition (linear re-composition) at the receiver antenna.
- Two echoes k and l, each with different delays, can be resolved when the difference of the delay is larger than the response time of the receiver system bandwidth, so $|\Delta\tau_k - \Delta\tau_l| > 1/B$.
- The received signal changes when the physical geometry changes.
- Signals with different delays are uncorrelated, thus the channel can be treated as wide-sense stationary with uncorrelated scatterers (WSSUS). The following items outline the constraints placed on the model, once the channel is assumed to be WSSUS [16]:
  - The channel can be modeled as a multi-tap delay line filter, and the channel can be modeled with a discrete-time impulse response, with complex-valued echoes.
  - Phases are uniformly distributed throughout the unit circle.
  - The channel impulse response consists of three elements: the direct path, the near-echo region, and the far-echo region. The direct path model is dependent on whether or not the signal is subject to shadowing. In LOS conditions, a Rician distribution is used, whereas in shadowed conditions, a Rayleigh distribution with a log-normal-distributed mean power is used. In the near-echo region, the amplitude of the echoes will vary according

to a Rayleigh distribution, while the number of near echoes is Poisson-distributed. The delay distribution of the near echoes follows the exponential distribution. In the region of far echoes, the number of far echoes is Poisson distributed, and the amplitudes follow a Rayleigh distribution.

Further findings of the DLR measurement campaign are outlined in [68] - [72]. In [68] to [70], findings resulting from the measurement campaign are outlined. In this measurement campaign, a Zeppelin was used as the transmitting platform to simulate a GNSS satellite. This Zeppelin transmitted known signals with carrier frequencies between 1460 and 1560 MHz. A van was equipped with receiver hardware (to include a commercial wireless network analyzer to perform measurements), the Zeppelin and van were moved through various environments, and the data was processed to uncover the impact of the local environment on the signal. In [68] and [69] in particular, results of measurements performed by people for pedestrian-level measurements are provided as well. Initial findings regarding delays in signal reception at various elevation angles of the Zeppelin relative to the horizon are provided in [68]. Approximate signal delays between 4 and 10 μsec were experienced at elevation angles from 80° down to 10°. Further findings are presented in [69], after performing a statistical analysis of the data from the measurement campaign. This statistical analysis makes use of the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm, which enables data analysis at a higher resolution that what is available in [68]. A significant discovery is that interference from a transmitter at 80° has a higher impact in the urban environment than transmissions from lower elevations. It is asserted in [68] that this results from more powerful reflected signals radiating down to the receive antenna after being reflected off of buildings (possible specular reflection). These reflections do not exhibit as a large a delay (relative to the direct path signal) as the multipath observed when the transmitter is at a lower elevation, which will impact the GNSS code delay loop accumulation and discrimination functions to a greater degree than signals that are delayed longer. Findings in [70] include the impact of edges, corners, and exterior walls of buildings, trees, and lampposts over time on the signal. These findings were then compared with previously existing propagation models. These results were then used to construct a simulated environment, which in turn yields simulated received signals [71]. The environment makes use of pseudoranges to construct this simulated environment to account for data collection vehicle dynamics. The model used by DLR in construction of the signal simulator considers both multipath effects and receiver movement. The fundamental measurement obtained in the campaign is the channel impulse response $h(\Delta\tau, t)$ relative to the LOS signal transmitted by the Zeppelin. $h(\Delta\tau, t)$ is as follows [71]:

$$h(\Delta\tau, t) = \sum_{i=0}^{N} a_{r,i}(t) e^{-j\phi_{r,i}(t)} \delta\left(\Delta\tau - \Delta\tau_i(t)\right) \tag{2.33}$$

where there are N discrete multipath signal echoes received and $\phi_{r,i}(t)$ is the phase variation of the multipath signal (which depends on the receiver movement and the variations in the multipath environment over time). The attenuation of the multipath echo relative to the direct signal path $a_{r,i}(t)$ is equal to [71]:

$$a_{r,i}(t) = \frac{a_i(t)}{a_0(t)} \tag{2.34}$$

The relative excess path delay $\Delta\tau_i(t)$ is as follows [71]:

$$\Delta\tau_i(t) = \tau_i(t) - \tau_0(t) \tag{2.35}$$

For the work presented in [71], an impulse response calculation rate of 1 kHz is used, allowing a maximum mobile data collection vehicle speed of 100 m/s with respect to stationary reflectors. The start and stop time of the discrete number of echoes is recognized automatically using the output of the ESPRIT data processing algorithm [71]. Therefore, the number of echoes N is determined through this process as well as the delay range and amplitude of the echo. The maximum number of echoes used in the simulation in [71] is approximately 80, as echoes would be considered if they exhibit a power greater than 100 dB less than the power of the LOS signal.

The environment was constructed probabilistically using what would be expected for the population of various scatterers and reflectors in the environment. This model can then be used to simulate what would be expected in a given environment, leveraging electromagnetic propagation models to artificially generate a received GNSS signal. The findings uncovered during the measurement campaign are extended in [72] by providing likelihood distributions of reflections over time and space, from the perspective of pedestrians in city or suburban streets. These likelihood distributions are offered in contrast to the use of ray tracing algorithms typically found in literature on multipath propagation (e.g., [73]).

The DLR simulation was expanded further by integrating a particle filter-based algorithm for tracking the delay resulting from multipath, among other multipath estimation techniques [74]. Discussion of multipath in both the land mobile environment and the aeronautical environment is offered in [71]. The use of the particle filter provides an estimate that can then be used to mitigate multipath.

While the urban environment modeled by DLR is emphasized in the research reported in this dissertation, indoor navigation is another scenario to be considered in multipath modeling. Researchers at the University of Calgary are actively considering the difficulties in indoor navigation using GNSS signals. Part of this effort involves characterizing indoor multipath in both time and space, given various environments [75]. Several points are made in [75] when describing the impact of the indoor environment on GNSS receiver signal processing. First, long coherent integration times are used to counter signal attenuation and fading. Second, antenna motion during these longer coherent integration times decorrelate the received GNSS signal, which will reduce the gain offered by the coherent integration. On the other hand, there is a diversity gain offered through antenna motion, which improves signal detection. Given these conditions, [75] describes the process and results of measurements of correlation coefficients for static and moving antennas in various indoor multipath fading channels. Particular outcomes in this research are that in a structure such as a wooden house, indoor multipath appears to be Rice distributed, since there is a substantial line of sight GNSS signal available, depending on the transmitting satellite position relative to the indoor structure [75]. For a more substantial structure, such as a spacious lab area inside a concrete building, the Rayleigh distribution appears to more appropriately describe the multipath environment, since there is a negligible line of sight signal present [75]. Put differently, the correlation coefficient is stronger for a wooden house than for a concrete laboratory or workshop structure.

2.5 GNSS Signal Receiver Processing for High-Fidelity Simulation

2.5.1 Simulated Annealing. Simulated annealing (SAN) is one of several pattern classification techniques that are in widespread use to solve various stochastic search and optimization problems. In the context of the research presented in this dissertation, SAN is used in the algorithm outlined in Chapter III for obtaining parameters instances of multipath propagated signals. Stochastic search methods are designed to find parameters for constructing complicated models involving random behavior, without exhaustively searching through a solution space. Stochastic search methods are sometimes inspired by physical and biological processes (such as SAN and genetic algorithms, respectively). Unlike classification methods such as the use of linear discriminant functions or neural networks, stochastic methods such as SAN and evolutionary methods can be used in problems where modeling would be extremely complicated, as in the case of models best expressed with high dimensionality [76]. Another critical distinction between SAN and other stochastic search techniques is the willingness built within the SAN algorithm to give up lower energy states presented by local minima in temporary favor of higher energy states [77]. Many other optimization techniques behave greedily, as defined in optimization literature [76]. Greedy techniques always pursue lower energy states. This may lead to a rapid decrease in energy state, but may also lead to the descent into a local minimum that is not the global minimum. Because SAN will temporarily surrender lower energy states in favor of higher energy states, the quick gain of a rapid decrease into a local minimum is avoided. Because this is the case, SAN will typically require significant computational effort in order to obtain convergence upon a global minimum. The tradeoff is the greater assurance of achieving convergence upon the global minimum than in gradient search algorithms. In SAN, local minima are avoided by controlling the magnitude reduction of the random perturbations used in performing the scouring of the search space posed by the parameters being considered [77]. The SAN algorithm is designed to travel through the local minima presented by a search space in search of a global minimum. An analogy to the simulated annealing process that may be useful to the layperson [78] is this: the surface of a pie pan is dented, leaving an uneven surface behind. A marble is then placed in the pie pan and the pan is shook gently. The marble will travel from local minimum to local minimum. With enough shaking of the pie pan, and the presence of a well-defined global minimum, the marble will eventually descend into the global minimum. Over time, the vigor of the shaking of the pie pan is reduced, so as to avoid disturbing the marble once it descends into the global minimum. Duda uses a physical analogy to illustrate how stochastic search works [76]. Given N physical magnets, each of which are indexed by i, and each of which can have either its North ($s_i = +1$) or South ($s_i = -1$) pole facing up, the function of the physical separation between magnets is w_{ij}, where j is the index of a magnet not numbered i. Each pair of magnets has an interaction energy E_{ij}. The sum of all energies is E. This sum of energies is expressed as follows [76]:

$$E = -\frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} w_{ij} s_i s_j \tag{2.36}$$

The task in this process is to find the configuration of states of the magnets that is most stable, therefore the configuration that has the lowest energy. It is readily obvious in this analogy that for large values of N, this problem would be very difficult to solve through an exhaustive search, as would be the case for 1 trillion magnets, for example. SAN provides a means by which this problem can be solved without requiring an exhaustive search. Annealing is the physical process by which magnets or atoms in an alloy reach a low-energy configuration. Heat is applied to the system (which increases the energy of the system), and then the temperature is gradually lowered, so the system relaxes into a low-energy configuration. As the temperature lowers, the probability of the system reaching an optimum configuration is increased [76]. There is the possibility that a local, yet not global, minimum is reached (compared to a golf hole in a golf course in [76]). The magnet analogy presented by Duda can be further complicated in cases where the magnets are interconnected by weights, therefore the energy associated with the magnet's direction is dictated by the states of other magnets as well as the factors previously mentioned [76].

To simulate annealing in the generalized stochastic search algorithm, the initial states of the system are randomized [76]. A high initial control parameter, the temperature T, is then chosen. A node i is then chosen at random, so its state s_i is equal to $+1$ (using the magnet analogy). The system energy E_a in this configuration is then calculated. The energy E_b for a new state $s_i = -1$ is then calculated. If $E_b < E_a$, then this change in state is accepted. If, on the other hand, $E_b > E_a$, E_b is accepted with a probability of $e^{-\Delta E_{ab}/T}$ [76], where $\Delta E_{ab} = E_b - E_a$. In contrast to other pattern classification algorithms, where the lower energy state is always accepted (as in naive gradient descent), SAN requires that the *higher* energy state is sometimes accepted. This acceptance of higher energy state is what allows the algorithm to move out of unacceptable local minima that would falsely be declared an optimal value. The process of selecting and testing states, referred to as polling, continues for several iterations. The temperature parameter T is then lowered in value, and polling is continued. Polling is continued for each node until each node has been acted upon for several iterations. The temperature is lowered again, and the cycle continues until T is near zero. If T is lowered slowly enough, then the system most likely (but not certainly) will be in a low-energy state, which in the annealing analogy means a global energy minimum. Note that the difference in energies between states is what is considered in the algorithm. This means that only nodes connected to the node being polled are considered. The nodes not connected to the polled node remain in the same state, to be addressed later as the algorithm leaps from node to node. Duda provides a pseudocode algorithm that can be used in writing computer code for the SAN algorithm [76]. Parameters that must be considered include the starting temperature, the ending temperature, the rate of temperature decrease, and the stopping criterion. The starting temperature, $T(1)$ must be high enough that all configurations have approximately the same probability (so $T(1)$ must be higher than the maximum energy difference shared between any configurations). The cooling rate must be gradual, to avoid trapping the system in a suboptimal local minimum. The final temperature must be low enough that there is very little chance that the system could move out of a globally optimal state. The temperature decrease rate is typically selected to be proportional, so the following equation would be used to express the temperature cooling rate [76]:

$$T(k+1) = cT(k) \tag{2.37}$$

where $0 < c < 1$. Duda states that typical value of c range from 0.8 to 0.99. This parameter will dictate the rate at which the algorithm works, but SAN algorithms are inherently slow to converge to a minimal solution.

Spall presents a variation on the description of the algorithm as well [77]. Duda's algorithm description is provided here to retain the notation used in the magnet analogy.

No previous work in the area of SAN in the modeling of GNSS channels has been uncovered. The following section outlines previous work in the area of SAN in modeling wireless communications channels.

The structure of GNSS signals is no different than what might be found in a Code Division Multiple Access (CDMA) or Frequency Division Multiplexed Binary Phase Shift Keying (FDM-BPSK) digital signal used in a communications link. Therefore, any work that has been performed in the past on the modeling or simulation of these kinds of communications signals that makes use of SAN is applicable to the research proposed here. There are several different publications that describe how

SAN is used in wireless communications. The most prevalent use is in the area of dynamic channel (or frequency band) assignment [79], proposed potentially as part of a cognitive radio infrastructure. SAN has also been used in an algorithm designed to estimate and track the channel state of a Rayleigh fading multipath channel [80]. This work does not, however, make use of SAN in the determination of a optimal channel description, but instead uses SAN in the computation of state space noise (the **G** matrix in Maybeck's state space notation [81]). While this use of SAN appears to be novel, this does not make use of the SAN algorithm for the same particular problem to be addressed in the research outlined in this dissertation. SAN has also been used in previous work to train a hidden Markov model (making use of eight states) used to model burst error sources in a wireless channel [82]. This differs from the research proposed in this dissertation, in that there is no use of hidden Markov models proposed in this dissertation. SAN has also been used in training a neural network used to model fading multipath in a wireless communications channel [83]. Again, this contrasts with the research proposed in this dissertation, in that there is no use of a neural network proposed here. SAN has also been used in modeling a system in generalized wireless sensor nodes [84]. This article, however, in intended to introduce a parallelized approach to SAN that makes use of the sensor network only as a test case with which to demonstrate the effectiveness of the algorithm. SAN has also been used to optimize the beamforming algorithm in a Multiple-Input/Multiple-Output (MIMO) communications channel [85]. Though [85] provides a good explanation of the SAN algorithm, this literature does not directly apply to the research proposed in this dissertation.

Literature is available that outlines the use of simulated annealing to refine GNSS-based position estimates, using both carrier- and code-phase (pseudorange) measurements [86] - [88].

2.6 Expectation Maximization (EM) and the Space Alternating Generalized Expectation (SAGE) Maximization Algorithms

This section outlines the theory behind the algorithm that most closely compares with the SDPA algorithm developed for the research presented in this dissertation. The most comparable algorithm is the SAGE algorithm. The theory behind the Expectation Maximization (EM) algorithm for which SAGE provides a generalized approximation is discussed in Section 2.6.1. The theory behind the SAGE algorithm itself is discussed in Section 2.6.2. The implementation of the SAGE algorithm to decompose received GNSS signals is outlined in Section 2.6.3.

2.6.1 Expectation Maximization (EM). The EM algorithm provides a way to iteratively compute maximum likelihood (ML) estimates [89]. The EM algorithm is particularly useful in what are called "incomplete data problems"—problems in which estimation is complicated by the absence of some component of the data (so estimation can't be completed until the incomplete data is somehow determined). There are two steps to the EM algorithm, the Expectation step (E-step) and the Maximization step (M-step). A compact explanation of the EM algorithm is provided in [90]. This explanation has been rewritten to apply to the scenario addressed in this dissertation. An observed data vector $\mathbf{r}[n]$ is expressed as follows:

$$\mathbf{r}[n] = \sum_{m=0}^{M-1} \{\mathbf{s}_m[n]\} + \boldsymbol{\eta}[n] \qquad (2.38)$$

where n provides a discrete time (sample) index and there are M signals $\mathbf{s}_m[n]$ indexed using m that are summed with additive noise $\boldsymbol{\eta}[n]$. Each of the signals $\mathbf{s}_m[n]$ has an associated set of parameters $\boldsymbol{\theta}_m$ (such as the signal amplitude A_m, Doppler frequency f_m, relative propagation delay τ_m, and initial carrier phase ϕ_m) that are used to construct $\mathbf{s}_m[n]$. If parameters are available for all the signals $\mathbf{s}_m[n]$ present within $\mathbf{r}[n]$ except for one, the signal for which parameters aren't available is indexed as p. Given the association between signals $\mathbf{s}_m[n]$ and parameters $\boldsymbol{\theta}_m$, $\mathbf{s}_m[n]$ can alternately be described as $\mathbf{s}_m(\boldsymbol{\theta}_m)$. Given this notation, (2.38) can be rewritten as follows:

$$\mathbf{s}_p(\boldsymbol{\theta}_p) = \mathbf{r}[n] - \boldsymbol{\eta}[n] - \sum_{\substack{m=0 \\ m \neq p}}^{M-1} \{\mathbf{s}_m(\boldsymbol{\theta}_m)\} \tag{2.39}$$

If the presence of noise $\boldsymbol{\eta}[n]$ in (2.39) is neglected, the E-step for this scenario is expressed as follows [89], [90]:

$$\mathbf{s}_p(\boldsymbol{\theta}_p) \approx \mathbf{r}[n] - \sum_{\substack{m=0 \\ m \neq p}}^{M-1} \{\mathbf{s}_m(\boldsymbol{\theta}_m)\} \tag{2.40}$$

The complete signal data (which includes the signal with parameters $\mathbf{s}_p(\boldsymbol{\theta}_p)$) is defined as follows:

$$\mathbf{y} = \sum_{m=0}^{M-1} \{\mathbf{s}_m(\boldsymbol{\theta}_m)\} \tag{2.41}$$

The M-step is then defined as follows [90]:

$$\hat{\boldsymbol{\theta}}_p = \arg\max_{\boldsymbol{\theta}} \left[p\left(\mathbf{y}|\boldsymbol{\theta}\right)\right] \tag{2.42}$$

$\hat{\boldsymbol{\theta}}_p$ is therefore the maximum likelihood estimate of $\boldsymbol{\theta}$ given $\mathbf{y}$. The EM algorithm iterates between signals indexed from $m = 0$ to $M - 1$, obtaining estimates for $\boldsymbol{\theta}_m$ through the E-step and M-step until the parameters settle at converged estimate values.

The EM algorithm has been shown in literature to be effective in solving for parameters in scenarios constructed as expressed in (2.38), where signals $\mathbf{s}_m[n]$ additionally originate from different spatial direction (azimuth, for example) [91], [92]. This research permits the means by which multipath can be mitigated through the use of beamforming, and is further extended in [93] to obtain parameter values for signal amplitudes and propagation delays.

2.6.2 Space Alternating Generalized Expectation Maximization (SAGE). The SAGE algorithm modifies the EM algorithm to solve sequentially for individual parameters in $\boldsymbol{\theta}$. This simplifies the M-step and generally improves the convergence rate over what is expected from the EM algorithm [94]. The term "hidden data space" used in this dissertation to describe signal decomposition using the SAGE algorithm in Section 2.6.3 is introduced in [94]. Multiple hidden data spaces are used in SAGE to generalize the complete data concept used in the EM algorithm [95]. The hidden data spaces can then be freely ordered by the algorithm user for the most suitable sequence. Because of the level of detail presented in Section 2.6.3, a summary of previously published literature on the subject of SAGE as it applies to multipath ray waveform parameter estimation will be provided in this section. The mechanics of the SAGE algorithm, to include equations, will be presented in Section 2.6.3. Where appropriate, sources for the content discussed in this section and applied in Section 2.6.3 will be cited with the subprocess or equation.

The SAGE algorithm is introduced in [94], but the initial work to outline the practical applicability of SAGE to the problem discussed in Section 2.6.1 is presented in [96] - [98]. The use of SAGE in the practical estimation of signal parameters such as the azimuthal incidence angle, the complex amplitude (magnitude and initial carrier phase), and the propagation delay is introduced in [96]. Signal parameter estimation using SAGE is extended in [97] to include estimation of Doppler frequency as well as the previously mentioned parameters. In [97], the order of estimation of parameters for each signal $\mathbf{s}_m(\boldsymbol{\theta}_m)$ in the case when only one antenna aperture is used in signal reception is as follows: τ_m, α_m, f_m, and then α_m again. SAGE literature uses complex amplitudes expressed as follows: $\alpha_m = A_m \exp{(j\phi_m)}$. The resolution and convergence speed of the SAGE algorithm are considered in detail in [99].

The first known consideration of SAGE in the estimation of GNSS signal multipath propagation delays is provided in [57]. The ingredients required to apply SAGE to GNSS multipath delay estimation are outlined in [57], and are clearly modeled heavily after [98]. Propagation delay estimation performance results from simulations are provided in [57]. Also considered in [57] is the initialization of the SAGE algorithm. In this paper, all parameters for every ray waveforms present in the received signal are initially equal to zero, though there is mention of the use of the Multiple Signal Classification (MUSIC) algorithm [100] to obtain initial estimates of the propagation delays for each of the ray waveforms. The number of ray waveforms present in the model of the received signal is explicitly treated as given.

The integration of SAGE-based parameter estimation techniques into a software GNSS receiver for use in the estimation of propagation delays present in simulated signals is outlined in [101]. This work is expanded upon in [102] through the integration of SAGE-based parameter estimation into a multi-branch delay lock loop (DLL) software GNSS receiver, similarly to MEDLL. The intent of [102] is to compare multiple parameter estimators within a framework with shared receiver structure surrounding the estimators.

The use of SAGE in conjunction with adaptive array processing is considered in [103]. This fusion of techniques works to further mitigate multipath. SAGE in conjunction with a multi-aperture array is shown to reduce multipath effects in GNSS signals, which follows from early consideration of EM and SAGE to estimate the angle of arrival of received signals [91] [92]. References [104] and [105] are similar to [103], in that both publications discuss multipath mitigation using an antenna array and SAGE in estimation of the direction of arrival of received ray waveforms. Since the research documented in this dissertation considers the single receiver aperture scenario, these papers provide background into the application of SAGE to the problem of GNSS multipath, but do not directly apply to the problem addressed in this document.

2.6.3 Signal Decomposition using the SAGE Algorithm.

EM and SAGE represent what is believed to be the state-of-the-art in the estimation of GNSS multipath parameters. This section outlines how a SAGE-based decomposition process is constructed, making use of previously published papers and community-developed resources that are available to the general public. The sequence in which parameters are estimated originates directly from [97], thus estimation of the propagation delay precludes estimation of the Doppler frequency. Estimation of the Doppler frequency does not require prior knowledge of the complex amplitude for the waveform in question.

The SAGE algorithm can be embedded within an iterative decomposition process by which a received data vector $\mathbf{r}_p$ is decomposed to obtain parameters for direct path and multipath ray waveforms. The duration of the received data vector is referred to as the pre-detection integration time (PIT). The PIT is indexed using p, so successive vectors of received data are indexed successively ($p \in \{0, \mathbb{N}\}$). When received signals are described, m is used to index the direct path and multipath ray waveform estimates, where M is the total number of multipath rays. Ray waveform parameter estimates are obtained iteratively, so precise notation to describe waveform parameter estimates must not only provide information on the waveform estimate m, but the iteration that contains m, as the same signal content is addressed repeatedly in multiple iterations. To alleviate this, ε is used to index the set of ray waveform parameter estimates output in iteration ε, referred to in this document as an "ensemble." ε is equal to zero for the first ensemble. The first ensemble of estimates will contain one ray. The second ensemble of estimates will contain two rays. The third ensemble of estimates will contain three rays, and so on. An example of this notation would be the communication of the second multipath ray ($m = 2$) from the fourth ensemble obtained using multiray processing ($\varepsilon = 3$) of the tenth PIT, $p = 9$. This multipath ray waveform estimate would be expressed as $\hat{\mathbf{r}}_{p_{\varepsilon_m}} = \hat{\mathbf{r}}_{9_{3_2}}$. There may be instances using this notation where the ensemble at large is considered, rather than individual rays within an ensemble. When this is the case, notation only includes the PIT index p and the ensemble index ε. An example of this would be $\hat{\mathbf{r}}_{p_\varepsilon}$.

The SAGE-based decomposition process accepts either of the following sets of initial ray waveform parameter estimates as inputs [57]:

- When the ray waveform estimate ensemble index $\varepsilon = 0$:

- The initial estimate of the ray waveform amplitude $\tilde{A}_{p_\varepsilon}$
- The initial estimate of the ray waveform Doppler frequency $\tilde{f}_{p_\varepsilon}$
- The initial estimate of the ray waveform propagation delay offset $\tilde{\tau}_{p_\varepsilon}$
- The initial estimate of the ray waveform carrier phase $\tilde{\phi}_{p_\varepsilon}$

- When $\varepsilon > 0$:
  - The vector of initial estimates of amplitudes for the ensemble of ray waveforms $\tilde{\mathbf{A}}_{p_\varepsilon}$
  - The vector of initial estimates of Doppler frequencies for the ensemble of ray waveforms $\tilde{\mathbf{f}}_{p_\varepsilon}$
  - The vector of initial estimates of propagation delay offsets for the ensemble of ray waveforms $\tilde{\boldsymbol{\tau}}_{p_\varepsilon}$
  - The vector of initial estimates of carrier phases for the ensemble of ray waveforms $\tilde{\boldsymbol{\phi}}_{p_\varepsilon}$

Output from this process are the final ray ensemble parameter estimates associated with the estimate waveform $\hat{\mathbf{r}}_{p_\varepsilon}$ (which contains the summation of direct path and multipath ray waveforms, and neglects portions of the received data that are determined to be noise). The specific parameters output from this process are as follows: the peak amplitude estimates $\hat{\mathbf{A}}_{p_\varepsilon}$, the frequency estimate $\hat{\mathbf{f}}_{p_\varepsilon}$, the propagation delay estimates $\hat{\boldsymbol{\tau}}_{p_\varepsilon}$, and the carrier phase estimates $\hat{\boldsymbol{\phi}}_{p_\varepsilon}$. Estimates are obtained for each of the $\varepsilon + 1$ rays in ensemble ε.

GNSS literature makes use of the terms "search space" [106] or "search grid" [107] to describe what is really a subset of the short-time Fourier transform (STFT) obtained from received data. The operations in the STFT computation are modified to additionally despread the direct sequence spread spectrum encoded GNSS signal content. The search space contains the region of the STFT matrix that is likely to contain GNSS signal content. The terms "search space" and "search grid" can both be found in literature to describe the same STFT matrix output. In this dissertation, the terms "search space" and "search grid" are both used to describe the STFT matrix and its constituent signal and noise content.

Before discussing the search space decomposition process, $\tilde{\mathbf{A}}_{p_\varepsilon}$, $\tilde{\mathbf{f}}_{p_\varepsilon}$, $\tilde{\boldsymbol{\tau}}_{p_\varepsilon}$, and $\tilde{\boldsymbol{\phi}}_{p_\varepsilon}$ are defined as follows:

$$\tilde{\mathbf{A}}_{p_\varepsilon} = \left[\hat{\mathbf{A}}_{p_{\varepsilon-1}}, \tilde{A}_{p_\varepsilon}\right] \tag{2.43}$$

$$\tilde{\mathbf{f}}_{p_\varepsilon} = \left[\hat{\mathbf{f}}_{p_{\varepsilon-1}}, \tilde{f}_{p_\varepsilon}\right] \tag{2.44}$$

$$\tilde{\boldsymbol{\tau}}_{p_\varepsilon} = \left[\hat{\boldsymbol{\tau}}_{p_{\varepsilon-1}}, \tilde{\tau}_{p_\varepsilon}\right] \tag{2.45}$$

$$\tilde{\boldsymbol{\phi}}_{p_\varepsilon} = \left[\hat{\boldsymbol{\phi}}_{p_{\varepsilon-1}}, \tilde{\phi}_{p_\varepsilon}\right] \tag{2.46}$$

Figure 2.1 illustrates the flowchart outlining the process used to decompose the received data in PIT index p. This flowchart involves the execution of several subprocesses, corresponding with the items in this section, where discussion of the specifics of each of these subprocesses is provided. The specific implementation of the SAGE algorithm provided in this discussion is a derivative of the algorithm developed for frequency estimation and discussed as part of the Matlab Spectral Analysis and Linear Prediction toolbox mentioned in [108]. The subprocesses for the SAGE-based decomposition are as follows: defining of the boundaries for the ranges of trial parameter values (Section 2.6.3.1), subtraction of interfering ray waveforms from received data (Section 2.6.3.2), estimation of the propagation delay for the mth ray waveform (Section 2.6.3.3), estimation of the Doppler frequency for the mth ray waveform (Section 2.6.3.4), computation of the amplitude and initial carrier phase for the mth ray waveform (Section 2.6.3.5), and updating of the SAGE ray waveform parameter matrix (Section 2.6.3.6).

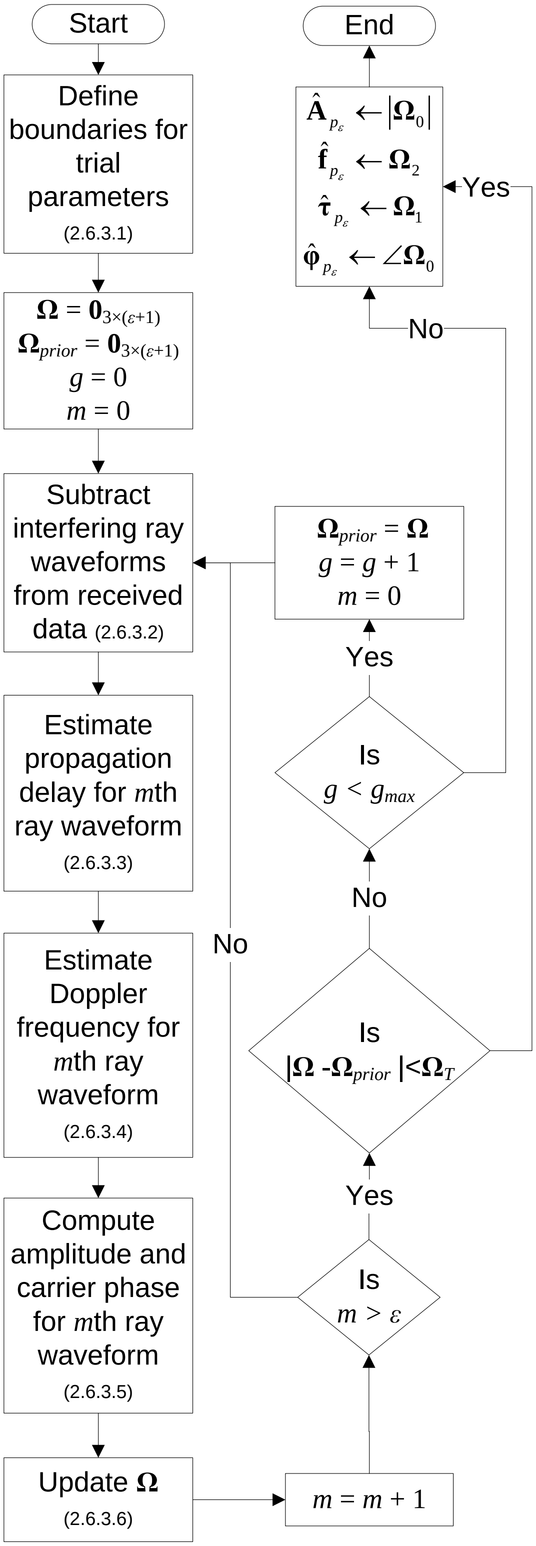
Start
Define boundaries for trial parameters
(2.6.3.1)
$\mathbf{\Omega} = \mathbf{0}_{3\times(\varepsilon+1)}$
$\mathbf{\Omega}_{prior} = \mathbf{0}_{3\times(\varepsilon+1)}$
$g = 0$
$m = 0$
Subtract interfering ray waveforms from received data (2.6.3.2)
Estimate propagation delay for mth ray waveform
(2.6.3.3)
Estimate Doppler frequency for mth ray waveform
(2.6.3.4)
Compute amplitude and carrier phase for mth ray waveform
(2.6.3.5)
Update $\mathbf{\Omega}$
(2.6.3.6)
$m = m + 1$
Is $m > \varepsilon$
No
Yes
Is $|\mathbf{\Omega} - \mathbf{\Omega}_{prior}| < \mathbf{\Omega}_T$
Yes
No
Is $g < g_{max}$
Yes
No
$\mathbf{\Omega}_{prior} = \mathbf{\Omega}$
$g = g + 1$
$m = 0$
$\hat{\mathbf{A}}_{p_\varepsilon} \leftarrow |\mathbf{\Omega}_0|$
$\hat{\mathbf{f}}_{p_\varepsilon} \leftarrow \mathbf{\Omega}_2$
$\hat{\boldsymbol{\tau}}_{p_\varepsilon} \leftarrow \mathbf{\Omega}_1$
$\hat{\boldsymbol{\varphi}}_{p_\varepsilon} \leftarrow \angle\mathbf{\Omega}_0$
End

Figure 2.1: Flowchart illustrating the SAGE-based decomposition process. Processes are outlined in this section, with exceptions in cases where the flowchart process label is self-explanatory, such as "$m = m + 1$." Each block is annotated with the section number where the subprocess in this flowchart is explained in depth.

2.6.3.1 Define Boundaries for Trial Parameters. This subprocess accepts either $\tilde{f}_{p0}$ and $\tilde{\tau}_{p0}$ (in cases when $\varepsilon = 0$), or $\tilde{\mathbf{f}}_{p0}$ and $\tilde{\boldsymbol{\tau}}_{p0}$ (in cases when $\varepsilon > 0$). Output from this process are boundary values (both minimum and maximum) for the trial parameters used to generate the trial ray waveforms being considered. Parameter boundaries are output for the carrier frequency ($f_{\min}$ and $f_{\max}$) and the propagation delay ($\tau_{\min}$ and $\tau_{\max}$). Also made available is the number of samples in $\mathbf{r}_p$, N_p.

This subprocess is used to simply define the ranges from which trial parameters for the carrier frequency $\mathbf{f}_{trial}$ and the propagation delay $\boldsymbol{\tau}_{trial}$ in the decomposition process may be obtained. These ranges are defined as follows:

$$\mathbf{f}_{trial} \in [f_{\min}, f_{\max}] \tag{2.47}$$

$$\boldsymbol{\tau}_{trial} \in [\tau_{\min}, \tau_{\max}] \tag{2.48}$$

where trial parameter vectors $\mathbf{f}_{trial}$ and $\boldsymbol{\tau}_{trial}$ are expressed as follows:

$$\mathbf{f}_{trial} = [f_{trial_m}]_{m=0}^{\varepsilon} \tag{2.49}$$

$$\boldsymbol{\tau}_{trial} = [\tau_{trial_m}]_{m=0}^{\varepsilon} \tag{2.50}$$

The carrier frequency trial parameter range boundaries are $f_{\min} = \tilde{f}_{p0} - 1.1/T_{PIT}$ and $f_{\max} = \tilde{f}_{p0} + 1.1/T_{PIT}$ (in cases when $\varepsilon = 0$) or $f_{\min} = \hat{\mathbf{f}}_{p0} - 1.1/T_{PIT}$ and $f_{\max} = \hat{\mathbf{f}}_{p0} + 1.1/T_{PIT}$ (in cases when $\varepsilon > 0$). T_{PIT} is the pre-detection integration period (in seconds). In cases when $\varepsilon > 0$, $f_{\min}$ and $f_{\max}$ are scalars, since $\hat{\mathbf{f}}_{p0} = \hat{f}_{p0_0}$ (only one ray waveform is parameterized in the estimate waveform generated from the initial ensemble, since the highest ray index for an ensemble is equal to the ensemble index itself). The propagation delay trial parameter range boundaries are $\tau_{\min} = \tilde{\tau}_{p0} - 1.1T_c$ and $\tau_{\max} = \tilde{\tau}_{p0} + 1.1T_c$ (in cases when $\varepsilon = 0$) or $\tau_{\min} = \hat{\boldsymbol{\tau}}_{p0} - 1.1T_c$ and $\tau_{\max} = \hat{\boldsymbol{\tau}}_{p0} + 1.1T_c$ (in cases when $\varepsilon > 0$). In cases when $\varepsilon > 0$, $\tau_{\min}$ and $\tau_{\max}$ are scalars, since $\hat{\boldsymbol{\tau}}_{p0} = \hat{\tau}_{p0_0}$.

Subsequently to this subprocess is the subprocess denoted "$\mathbf{\Omega} = \mathbf{0}_{3\times(\varepsilon+1)}$, $\mathbf{\Omega}_{\boldsymbol{prior}} = \mathbf{0}_{3\times(\varepsilon+1)}$, $g = 0$, $m = 0$". The following equations outline the operations in this subsequent subprocess, where $\mathbf{\Omega}$ and $\mathbf{\Omega}_{\boldsymbol{prior}}$ are initialized using zero matrices in accordance with [57]:

$$\mathbf{\Omega} = \mathbf{0}_{3\times(\varepsilon+1)} \tag{2.51}$$

$$\mathbf{\Omega}_{\boldsymbol{prior}} = \mathbf{0}_{3\times(\varepsilon+1)} \tag{2.52}$$

$$g = 0 \tag{2.53}$$

$$m = 0 \tag{2.54}$$

2.6.3.2 Subtraction of Interfering Ray Waveforms from Received Data.

This process accepts the SAGE ray waveform parameter matrix $\mathbf{\Omega}$ as an input and outputs the difference between the received integration period-sized data and the summation of estimate multipath ray waveforms that are not being considered in the current iteration m (which equals the ray index being considered), $\mathbf{r}_{HDS}$. $\mathbf{r}_{HDS}$ is annotated as such because it makes use of the hidden data space discussed in SAGE literature [94].

Ray waveform parameters must first be extracted from $\mathbf{\Omega}$. The following equations are used to extract these parameters:

$$\breve{\mathbf{A}}_{p_\varepsilon} = \left[\breve{A}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} = [|\mathbf{\Omega}(0,m)|]_{m=0}^{\varepsilon} \tag{2.55}$$

$$\breve{\mathbf{f}}_{p_\varepsilon} = \left[\breve{f}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} = [\mathbf{\Omega}(3,m)]_{m=0}^{\varepsilon} \tag{2.56}$$

$$\breve{\boldsymbol{\tau}}_{p_\varepsilon} = \left[\breve{\tau}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} = [\mathbf{\Omega}(2,m)]_{m=0}^{\varepsilon} \tag{2.57}$$

$$\breve{\boldsymbol{\phi}}_{p_\varepsilon} = \left[\breve{\phi}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} = [\angle\mathbf{\Omega}(0,m)]_{m=0}^{\varepsilon} \tag{2.58}$$

where $\angle$ denotes the phase angle computation, which is the same as the four quadrant arctangent operation $\arg(\mathbf{\Omega}(0,m))$. There are four row vectors in (2.55) - (2.58) denoted with the following notation: $[\mathbf{\Omega}(\bullet,m)]_{m=0}^{\varepsilon}$. These vectors are composed of ray waveform parameters values associated with each of the estimates obtained using the SAGE algorithm. The first element in the vector is from the waveform denoted as $m = 0$, the second element is from the waveform denoted as $m = 1$, and so on. There are a total of ε parameters in each vector, as there are $\varepsilon + 1$ ray waveforms that are parameterized after the εth iteration of the SAGE-based decomposition engine.

To perform this operation, a modification of the summation in (3.83) can be used to obtain the received signal with estimates of interfering signals subtracted, $\mathbf{r}_{HDS}$, as follows [57] [97]:

$$\mathbf{r}_{HDS} = \mathbf{r}_p - \sum_{\substack{\ell=0 \\ \ell \neq m}}^{\varepsilon} \left[\breve{\mathbf{r}}_{p_{\varepsilon_\ell}} \right] \tag{2.59}$$

where $\breve{\mathbf{r}}_{p_{\varepsilon_\ell}}$ is as follows:

$$\breve{\mathbf{r}}_{p_{\varepsilon_\ell}} = \begin{cases} \breve{A}_{p_{\varepsilon_\ell}} \breve{\mathbf{x}}_{p_{\varepsilon_\ell}} \circ \left[\exp\left(j \left(2\pi (f_{\mathrm{IF}} + \breve{f}_{p_{\varepsilon_\ell}}) T_s w + \breve{\phi}_{p_{\varepsilon_\ell}} \right) \right) \right]_{w=0}^{N_p - 1}, & \mathbf{r}_p \in \mathbb{C} \\ \mathrm{Re} \left\{ \breve{A}_{p_{\varepsilon_\ell}} \breve{\mathbf{x}}_{p_{\varepsilon_\ell}} \circ \left[\exp\left(j \left(2\pi (f_{\mathrm{IF}} + \breve{f}_{p_{\varepsilon_\ell}}) T_s w + \breve{\phi}_{p_{\varepsilon_\ell}} \right) \right) \right]_{w=0}^{N_p - 1} \right\}, & \mathbf{r}_p \in \mathbb{R} \end{cases} \tag{2.60}$$

where $\circ$ denotes Hadamard multiplication. The vectors $[\exp(\bullet)]_{w=0}^{N_p-1}$ serve as modulation vectors. These vectors are indexed using w and are of length N_p. The length N_p is the number of samples in a received data vector of duration equal to the integration period. The computation of $\breve{\mathbf{x}}_{p_{\varepsilon_\ell}}$ is performed using the following equation:

$$\breve{\mathbf{x}}_{p_{\varepsilon_\ell}} = \mathbf{x} \left(\breve{\tau}_{p_{\varepsilon_\ell}} \right) \tag{2.61}$$

where $\mathbf{x}$ is the direct sequence spread spectrum code vector for the PRN of interest that has been delayed by $\breve{\tau}_{p_{\varepsilon_\ell}}$.

2.6.3.3 Estimation of Propagation Delay for Ray Waveform of Interest. This process accepts $\mathbf{r}_{HDS}$, $\breve{f}_{p_{\varepsilon_m}}$, $\tau_{\min}$, and $\tau_{\max}$ as inputs and outputs the SAGE-based maximum likelihood estimate of the propagation delay, $\breve{\tau}_{p_{\varepsilon_m}}$ [57] [97] [98].

Figure 2.2 illustrates the subprocess of estimating the propagation delay contained within the SAGE-based decomposition process. This figure illustrates the operations outlined in this section.

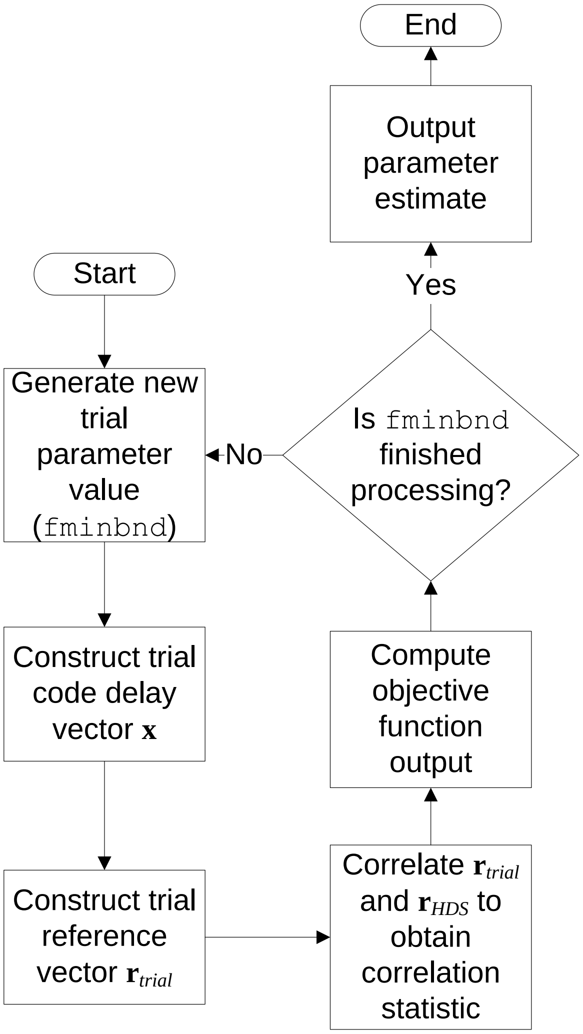
End
Output parameter estimate
Yes
Start
Generate new trial parameter value (fminbnd)
No
Is fminbnd finished processing?
Construct trial code delay vector $\mathbf{x}$
Compute objective function output
Construct trial reference vector $\mathbf{r}_{trial}$
Correlate $\mathbf{r}_{trial}$ and $\mathbf{r}_{HDS}$ to obtain correlation statistic

Figure 2.2: Flowchart illustrating the process of estimating parameters using the SAGE algorithm. Parameter estimation (to include trial parameter generation) is performed using the Matlab `fminbnd` function.

Trial values of the propagation delay, τ_{trial}, must be generated to execute the SAGE algorithm. Practical implementation of the SAGE algorithm to estimate propagation delay involves use of the Matlab `fminbnd` function [108]. The `fminbnd` function is based on "golden section search and parabolic interpolation" [109]. The `fminbnd` function solves for the minimum of the argument, so the objective function must involve an inverse operation to estimate the propagation delay associated with the maximum correlation function output (which is a function of the delay).

The trial code delay vector $\mathbf{x}(\tau_{trial})$ must next be constructed. This vector is used to encode the reference signal using the PRN sequence associated with the signal of interest (in this research GPS C/A-coded signals). $\mathbf{x}(\tau_{trial})$ is computed as follows:

$$\mathbf{x}(\tau_{trial}) = \left[b \left[\left\lfloor N_c \left(\frac{wT_s - \tau_{trial}}{N_c T_c} - \left\lfloor \frac{wT_s - \tau_{trial}}{N_c T_c} \right\rfloor \right) \right\rfloor \right] \right]_{w=0}^{N_p - 1}, b \in \{-1, +1\} \tag{2.62}$$

where $b[\bullet]$ is an element of the PRN sequence vector $\mathbf{b}$ designated by the natural number inside the brackets (where $\mathbf{b}$ is the direct sequence spread spectrum encoding chip sequence of length N_c), $\lfloor \bullet \rfloor$ indicates a floor rounding operation (rounding down to the nearest integer on the real number line less than the value contained in the modified brackets), N_c is the number of chips in the PRN sequence ($N_c = 1023$ for GPS C/A-coded signals), and T_c is the chip duration (so $T_c = 0.001/N_c = 0.001/1023$ seconds for GPS C/A-coded signals).

The trial reference vector $\mathbf{r}_{trial}$ must now be constructed. This vector serves as the reference against which $\mathbf{r}_{HDS}$ is correlated. $\mathbf{r}_{trial}$ is computed as follows:

$$\mathbf{r}_{trial} = \mathbf{x}(\tau_{trial}) \circ \left[\exp\left(-j2\pi \left(f_{\mathrm{IF}} + \breve{f}_{p_{\varepsilon_m}} \right) T_s w \right) \right]_{w=0}^{N_p - 1} \tag{2.63}$$

where T_s is the sample period and N_p is the number of samples in the integration period of interest.

The trial reference vector $\mathbf{r}_{trial}$ is then correlated with $\mathbf{r}_{HDS}$, yielding the correlation statistic $\mathcal{C}$ using the following operation:

$$\mathcal{C} = \left\{ \begin{array}{c} \frac{1}{N_p} \sum_{w=0}^{N_p - 1} \left\{ \mathbf{r}_{trial_w} \cdot \mathbf{r}_{HDS_w} \right\}, \mathbf{r}_{HDS} \in \mathbb{C} \\ \frac{2}{N_p} \sum_{w=0}^{N_p - 1} \left\{ \mathbf{r}_{trial_w} \cdot \mathbf{r}_{HDS_w} \right\}, \mathbf{r}_{HDS} \in \mathbb{R} \end{array} \right\} \tag{2.64}$$

where $\mathbf{r}_{trial} = [\mathbf{r}_{trial_w}]_{w=0}^{N_p - 1}$ and $\mathbf{r}_{HDS} = [\mathbf{r}_{HDS_w}]_{w=0}^{N_p - 1}$.

The square of the magnitude of the correlation statistic is then inverted to obtain the objective function output $\mathcal{E}\left(\tau_{trial}, \breve{f}_{p_{\varepsilon_m}}\right)$. The value of τ_{trial} that minimizes $\mathcal{E}\left(\tau_{trial}, \breve{f}_{p_{\varepsilon_m}}\right)$, as determined by the `fminbnd` function transparently to the user, is then asserted as the propagation delay estimate $\check{\tau}_{p_{\varepsilon_m}}$ if the criteria to signify completion of the processing with the `fminbnd` function is met. Otherwise, this subprocess is restarted using a different trial propagation delay value.

2.6.3.4 Estimation of Doppler Frequency for Ray Waveform of Interest.

This process accepts $\mathbf{r}_{HDS}$, $\check{\tau}_{p_{\varepsilon_m}}$, $f_{\min}$, and $f_{\max}$ as inputs and outputs the SAGE-based maximum likelihood estimate of the propagation delay, $\breve{f}_{p_{\varepsilon_m}}$ [98].

As with the estimation of propagation delay outlined in Section 2.6.3.3, Figure 2.2 illustrates the subprocess of estimating the Doppler frequency contained within the SAGE-based decomposition process. Figure 2.2 illustrates the operations outlined in this section.

Trial values of the Doppler frequency, f_{trial}, must be generated to execute the SAGE algorithm. As with estimation of the propagation delay, practical implementation of the SAGE algorithm to estimate the Doppler frequency involves use of the Matlab `fminbnd` function, as discussed in Section 2.6.3.3.

The trial code delay vector $\mathbf{x}(\check{\tau}_{p_{\varepsilon_m}})$ must next be constructed. $\mathbf{x}(\check{\tau}_{p_{\varepsilon_m}})$ is computed as follows:

$$\mathbf{x}(\check{\tau}_{p_{\varepsilon_m}}) = \left[b \left[\left\lfloor N_c \left(\frac{wT_s - \check{\tau}_{p_{\varepsilon_m}}}{N_c T_c} - \left\lfloor \frac{wT_s - \check{\tau}_{p_{\varepsilon_m}}}{N_c T_c} \right\rfloor \right) \right\rfloor \right] \right]_{w=0}^{N_p - 1}, b \in \{-1, +1\} \qquad (2.65)$$

The trial reference vector $\mathbf{r}_{trial}$ must now be constructed. $\mathbf{r}_{trial}$ is computed as follows:

$$\mathbf{r}_{trial} = \mathbf{x}(\check{\tau}_{p_{\varepsilon_m}}) \circ \left[\exp\left(-j2\pi\left(f_{\mathrm{IF}} + f_{trial}\right) T_s w\right)\right]_{w=0}^{N_p-1} \qquad (2.66)$$

The trial reference vector $\mathbf{r}_{trial}$ is then correlated with $\mathbf{r}_{HDS}$ using (2.66) and (2.64) directly.

The square of the magnitude of the correlation statistic is then inverted to obtain the objective function output $\mathcal{E}\left(f_{trial}, \check{\tau}_{p_{\varepsilon_m}}\right)$. The value of f_{trial} that minimizes $\mathcal{E}\left(f_{trial}, \check{\tau}_{p_{\varepsilon_m}}\right)$, as determined by the `fminbnd` function transparently to the user, is then asserted as the Doppler frequency estimate $\check{f}_{p_{\varepsilon_m}}$ if the criteria to signify completion of the processing with the `fminbnd` function is met. Otherwise, this subprocess is restarted using a different trial Doppler frequency value.

2.6.3.5 Compute Amplitude and Carrier Phase for Ray Waveform of Interest. This process accepts $\mathbf{r}_{HDS}$, $\check{f}_{p_{\varepsilon_m}}$, and $\check{\tau}_{p_{\varepsilon_m}}$ as inputs and outputs the maximum likelihood estimates of the ray waveform amplitude and initial carrier phase, $\check{A}_{p_{\varepsilon_m}}$ and $\check{\phi}_{p_{\varepsilon_m}}$ respectively.

To compute $\check{A}_{p_{\varepsilon_m}}$ and $\check{\phi}_{p_{\varepsilon_m}}$, $\mathbf{x}(\check{\tau}_{p_{\varepsilon_m}})$ must first be computed. $\mathbf{x}(\check{\tau}_{p_{\varepsilon_m}})$ is computed using (2.65) directly [57] [97] [98].

$\mathbf{r}_{trial}$ must then be computed using the following equation:

$$\mathbf{r}_{trial} = \mathbf{x}(\check{\tau}_{p_{\varepsilon_m}}) \circ \left[\exp\left(-j2\pi\left(f_{\mathrm{IF}} + \check{f}_{p_{\varepsilon_m}}\right)T_s w\right)\right]_{w=0}^{N_p-1} \tag{2.67}$$

The trial reference vector $\mathbf{r}_{trial}$ is then correlated with $\mathbf{r}_{HDS}$ using (2.66) and (2.64) directly to obtain $\mathcal{C}$.

Finally, $\check{A}_{p_{\varepsilon_m}}$ and $\check{\phi}_{p_{\varepsilon_m}}$ are obtained directly using the following equations:

$$\check{A}_{p_{\varepsilon_m}} = |\mathcal{C}| \tag{2.68}$$

$$\check{\phi}_{p_{\varepsilon_m}} = \angle\mathcal{C} \tag{2.69}$$

2.6.3.6 Update SAGE Ray Waveform Parameter Matrix. This process accepts $\check{A}_{p_{\varepsilon_m}}$, $\check{f}_{p_{\varepsilon_m}}$, $\check{\tau}_{p_{\varepsilon_m}}$, and $\check{\phi}_{p_{\varepsilon_m}}$ as inputs and outputs an update to the mth column of $\mathbf{\Omega}$, where the remaining columns remain unchanged. This update is conducted using the following three equations:

$$\mathbf{\Omega}(1,m) = \check{A}_{p_{\varepsilon_m}} \exp\left(j\check{\phi}_{p_{\varepsilon_m}}\right) \tag{2.70}$$

$$\mathbf{\Omega}(2,m) = \check{\tau}_{p_{\varepsilon_m}} \tag{2.71}$$

$$\mathbf{\Omega}(3,m) = \check{f}_{p_{\varepsilon_m}} \tag{2.72}$$

The process that follows, denoted "$m = m + 1$", is used simply to increment the index of the ray begin considered for evaluation when making use of the hidden data space. This operation is expressed as follows:

$$m = m + 1 \tag{2.73}$$

Upon completion of the incrementing of the ray index, a decision point is reached. If $m > \varepsilon$, parameter estimation iteration g for the ensemble of $\varepsilon + 1$ rays is complete, and another decision point is reached, "Is $|\mathbf{\Omega} - \mathbf{\Omega}_{prior}| < \mathbf{\Omega}_T$". If $m \leq \varepsilon$, there are rays remaining for iteration g for which parameters remain to be estimated. In this case, processing would continue with subtraction of interfering ray waveforms from the received data, in order to consider the next ray in the ensemble. This subprocess is outlined in Section 2.6.3.2.

The decision point "Is $|\mathbf{\Omega} - \mathbf{\Omega}_{prior}| < \mathbf{\Omega}_T$" is used to determine if the solution for $\mathbf{\Omega}$ has converged. If the difference for every element in $|\mathbf{\Omega} - \mathbf{\Omega}_{prior}|$ is less than $\mathbf{\Omega}_T$, $\mathbf{\Omega}$ is declared to have converged, and the algorithm is ready to complete processing for ensemble ε. Otherwise, the decision point annotated as "Is $g < g_{\max}$" awaits. The value for $\mathbf{\Omega}_T$ used in this research is as follows:

$$\mathbf{\Omega}_T = \begin{bmatrix} 0.1 \times \mathbf{1}_{3\times(\varepsilon+1)} \\ 0.1c \times \mathbf{1}_{3\times(\varepsilon+1)} \\ 0.1 \times \mathbf{1}_{3\times(\varepsilon+1)} \end{bmatrix} \tag{2.74}$$

where the first row is unitless, the second row is expressed in sec, and the third row is expressed in Hz. This composition of $\mathbf{\Omega}_T$ is selected to provide a reasonable assurance of convergence, but probably restricts SAGE-based processing to computational burden without a meaningful practical gain in estimation accuracy.

The decision point "Is $g < g_{\max}$" is used to determine if the maximum number of SAGE processing iterations g for ensemble ε has been reached. If $g < g_{\max}$, the maximum number of iterations has not yet been reached, and the subprocess annotated as "$\mathbf{\Omega}_{prior} = \mathbf{\Omega}$, $g = g + 1$, $m = 0$" awaits. If $g = g_{\max}$, the algorithm is ready to complete processing for ensemble ε.

The subprocess annotated as "$\mathbf{\Omega}_{prior} = \mathbf{\Omega}$, $g = g+1$, $m = 0$" is used to reset or increment the algorithm's housekeeping variables. When this subprocess is complete, processing would continue with subtraction of interfering ray waveforms from the received data, in order to consider the 0th ray in the next ensemble. This subprocess is outlined in Section 2.6.3.2.

When the algorithm is ready to complete processing, the subprocess annotated as "$\hat{\mathbf{A}}_{p_\varepsilon} \leftarrow |\mathbf{\Omega}_0|$, $\hat{\mathbf{f}}_{p_\varepsilon} \leftarrow \mathbf{\Omega}_2$, $\hat{\boldsymbol{\tau}}_{p_\varepsilon} \leftarrow \mathbf{\Omega}_1$, $\hat{\boldsymbol{\phi}}_{p_\varepsilon} \leftarrow \angle\mathbf{\Omega}_0$" is performed using the following equations:

$$\hat{\mathbf{A}}_{p_\varepsilon} = [|\mathbf{\Omega}(0,m)|]_{m=0}^{\varepsilon} \tag{2.75}$$

$$\hat{\mathbf{f}}_{p_\varepsilon} = [\mathbf{\Omega}(2,m)]_{m=0}^{\varepsilon} \tag{2.76}$$

$$\hat{\boldsymbol{\tau}}_{p_\varepsilon} = [\mathbf{\Omega}(1,m)]_{m=0}^{\varepsilon} \tag{2.77}$$

$$\hat{\boldsymbol{\phi}}_{p_\varepsilon} = [\angle\mathbf{\Omega}(0,m)]_{m=0}^{\varepsilon} \tag{2.78}$$

2.7 Multipath Estimating Delay Locked Loop (MEDLL)

The MEDLL receiver originally developed primarily by van Nee makes use of maximum likelihood estimation [110], which offers the optimal estimation approach in the absence of prior knowledge being available for Bayesian estimation. With MEDLL, pseudorange errors due to multipath are reduced significantly. To perform

MEDLL processing, the following log-likelihood cost function is used [110]:

$$\Gamma(\theta) = \int_{T_0} \left| r(t) - A_1 c(t-\tau_1)e^{j\phi_1} - \sum_{i=2}^{M} \hat{A}_i c(t-\hat{\tau}_i)e^{j\hat{\phi}_i} \right|^2 dt \tag{2.79}$$

where $\theta = (A_1, A_2, \ldots, A_M, \tau_1, \tau_2, \ldots, \tau_M, \phi_1, \phi_2, \ldots \phi_M)$ is the array of unknown parameters for which to be solved, T_0 is the range of integration (so the range between the start and end of the integration period), $r(t)$ is the received signal, A_i is the amplitude of the ith ray waveform, $c(t-\tau_i)$ is the value of the code vector at $t-\tau_i$ (where τ_i is the relative propagation delay of the ith ray waveform), ϕ_i is the initial carrier phase of the ith ray waveform, and M is the number of received rays. Maximum likelihood estimates are obtained by setting to zero the partial derivatives of $\Gamma(\theta)$ with respect to each parameter in θ. This operation yields a set of nonlinear equations which present the possibility of being difficult to solve. This is overcome through use of an overall approximation to the cross correlation functions, $R_{rc}(\tau)$, making use of a set of reference correlation functions that have various delays, phases, and amplitude values associated with them. $R_{rc}(\tau)$ is computed using the following equation [110]:

$$R_{rc}(\tau) = \sum_{i=0}^{M} R_i(\tau) \tag{2.80}$$

where $R_i(\tau)$ is the portion of $R_{rc}(\tau)$ associated with the ith ray waveform.

A parallel bank of correlators that compute $R_{rc}(\tau)$ for various values of τ is built into the MEDLL receiver architecture [110]. Cross-correlation values $R_{rc}(\tau)$ for the various values of τ are input into the MEDLL processor.

MEDLL originally incorporated an interference cancellation component [110], as with SAGE as described in Section 2.6.3.2.

MEDLL makes use of a bank of correlators set up within delay locked loops [54]. These loops are required by definition in order to describe MEDLL architecture. The values output from the bank of correlators are tracked over time in a delay locked loop architecture in order to track multipath parameters.

As stated previously in this section, MEDLL offers the optimal parameter estimation approach through the use of maximum likelihood estimation. This optimality is the case as defined in terms of estimation theory, given the use of maximum likelihood estimation [111]. MEDLL is indeed the conditionally optimal means by which to obtain parameter estimates, but MEDLL also by definition requires the use of tracking loops that make use of information obtained from integration periods previous to the period being considered for estimation. This use of previous tracking loop outputs presents a possible error source where inaccurate estimates of parameters from previous integration periods impact the current estimates. In short, MEDLL does not process integration periods independently from each other.

2.8 Signal Detection Theory

SDPA processing provides an ensemble of ray estimates to be used in modeling the received GNSS signal. This ensemble is to consist of ray estimates describing the GNSS signal in the absence of noise. In the scenarios considered in this research, where the received GNSS signal in the presence of multipath interference is readily observed in the received search grid, the first few ray estimates almost inevitably will be obtained from content within the search grid consisting largely from the GNSS signal itself, with noise not having a great deal of influence on the ray parameter estimates. As decomposition progresses and the GNSS and multipath interference signals within the search grid are deducted, noise will have greater influence over the ray estimates. Eventually, if the decomposition algorithm perfectly deducts the GNSS and multipath interference signals from the search grid, if the algorithm continues it will obtain ray parameter estimates that strictly describe noise alone. Within the construct of the algorithm, so long as one of the local maxima within the search grid window of interest yields initial estimates that lead to a reduction in the difference between the received search grid and the estimate search grid, the algorithm will continue to perform the decomposition process, regardless of whether the reduction in difference between search grids results from the presence of a valid signal or multipath interference, or from the presence of noise. Therefore, a criteria must be implemented to dictate to the algorithm when no further decomposition is to be performed. To implement this stopping criteria, signal detection theory (SDT) is considered.

Section 2.8.1 provides theoretic background on the subject of SDT. The applicability of SDT to a signal decomposition stopping criteria is considered in Section 2.8.2. Theory behind determining the number of multipath rays present in a received signal is provided in Section 2.8.3. Finally, a discussion of literature considering the application of a stopping criteria to a signal decomposition algorithm is provided in Section 2.8.4.

2.8.1 Signal Detection Theory (SDT) Background. The motivation for SDT is the need for autonomous decision making. Detection theory problems involve making correct decisions among a finite number of alternatives, or states [111] [112].

In SDT, the evidence variable is the independent, or input, variable used to dictate the output of a decision engine [113]. Within this discussion of SDT background, the evidence variable is denoted χ. There will be two possibilities, or states, that take place, for purposes of this discussion. These two states are denoted s and n (implying signal plus noise and just noise, respectively). Therefore, the conditional probability that the evidence variable is equal to χ given state s has occurred is denoted $P(\chi|s)$. The conditional probability that the evidence variable is equal to χ given state n has occurred is denoted $P(\chi|n)$. The likelihood ratio, $l(\chi)$, is expressed as follows [113]:

$$l(\chi) = \frac{P(\chi|s)}{P(\chi|n)} \tag{2.81}$$

Note that the evidence variable χ may be multi-dimensional in nature. If this is the case, the evidence variable is denoted as $\vec{\chi} = [\chi_0, \chi_1, \chi_2, \cdots]$.

The input variables, or “stimulus events” $\mathscr{s}$ and $\mathscr{n}$ correspond with possible “response events” $\mathscr{S}$ and $\mathscr{N}$, respectively [113]. Alternately, the events can be expressed using a hypothesis testing convention. In this case, $\mathscr{S}$ and $\mathscr{N}$ are expressed as H_1 (detection) and H_0 (no detection), respectively [111].

There are four combinations of stimulus and response events [76], [113]: a “hit” (the decision engine correctly outputs a detection), a “miss” (the decision engine incorrectly outputs no detection), a “false alarm” (the decision engine incorrectly outputs a detection), and a “correct rejection” (the decision engine correctly outputs no detection). The probability of a “hit” is denoted by $P\left(\mathscr{S}|\mathscr{s}\right)$, the probability of a “miss” is denoted by $P\left(\mathscr{N}|\mathscr{s}\right)$, the probability of a “false alarm” is denoted by $P\left(\mathscr{S}|\mathscr{n}\right)$, and the probability of a “correct rejection” is denoted by $P\left(\mathscr{N}|\mathscr{n}\right)$. Using the hypothesis testing convention, a false alarm is called a Type I error, when H_1 is decided when H_0 is true. A miss is called a Type II error, when H_0 is decided when H_1 is true.

The decision rule used to decide between $\mathscr{S}$ and $\mathscr{N}$ is stated in terms of likelihood ratios as follows [113]:

$$\text{If } l\left(\mathscr{x}\right) < \mathscr{B}, \text{decide } \mathscr{N}; \text{else if } l\left(\mathscr{x}\right) \geq \mathscr{B}, \text{decide } \mathscr{S} \tag{2.82}$$

where the “decision criterion” $\mathscr{B}$ is valued at $\mathscr{B} = 1$, establishing a threshold for the decision engine whereby any value falling below $\mathscr{B}$ is decided not to be a detection, and any value falling at or above $\mathscr{B}$ is decided to be a detection. Note that decision rules may require that different regions of values of $\mathscr{x}$ result in the same decision [76].

The objective in establishing the “decision criterion” is to optimize the performance of the decision engine. When the prior probability $P\left(\mathscr{s}\right)$ does not equal the prior probability $P\left(\mathscr{n}\right)$, the “decision criterion” is expressed as follows [113]:

$$\mathscr{B} = \frac{P\left(\mathscr{n}\right)}{P\left(\mathscr{s}\right)} \tag{2.83}$$

In cases where the evidence variable is multi-dimensional, joint probabilities are used to establish decision criteria [111].

The decision rule can be stated differently as follows [76]:

$$\text{If } P\left(\mathscr{n}|\mathscr{x}\right) > P\left(\mathscr{s}|\mathscr{x}\right), \text{decide } \mathscr{N}; \text{else decide } \mathscr{S} \tag{2.84}$$

Until this point, the assumption is made in this discussion that no rewards or penalties are assigned to the various responses (for example, the cost of a false alarm being of higher consequence than the value of making a hit). To account for this possibility, the "decision criterion" is expressed as follows [113]:

$$\mathscr{B} = \frac{\left(V_{\mathscr{n}\mathscr{N}} + C_{\mathscr{n}\mathscr{S}}\right) P\left(\mathscr{n}\right)}{\left(V_{\mathscr{s}\mathscr{S}} + C_{\mathscr{s}\mathscr{N}}\right) P\left(\mathscr{s}\right)} \tag{2.85}$$

where $V_{\mathscr{n}\mathscr{N}}$ is the value of a correct rejection, $C_{\mathscr{n}\mathscr{S}}$ is the cost of a false alarm, $V_{\mathscr{s}\mathscr{S}}$ is the value of a hit, and $C_{\mathscr{s}\mathscr{N}}$ is the cost of a miss. Typically, rules are established to keep the proportion of false alarms below some sort of threshold. The tradeoff is that with a reduction in false alarms comes a reduction in the proportion of properly detected hits as well. Fundamentally, the decision engine should be designed to minimize risk (cost minus value) for every value of $\mathscr{x}$ [76].

Given the decision criterion in (2.85), the decision rule in (2.82) can be restated using the hypothesis testing convention to express the likelihood ratio test as follows [111]:

$$l\left(\mathscr{x}\right) \underset{\mathscr{N}}{\overset{\mathscr{S}}{\gtrless}} \mathscr{B} \tag{2.86}$$

How well the decision engine is able to make correct decisions while avoiding incorrect decisions is called sensitivity [113] (or discriminability [76]). The extent to which the decision engine favors one hypothesis over another is called bias.

Given the probability of a hit $(P(\mathcal{S}|\mathscr{s}))$ and the probability of a "false alarm" $(P(\mathcal{S}|\mathscr{n}))$ for various decision criteria $\mathscr{B}$, the receiver operating characteristic (ROC) curve can be generated [76] [113]. The ROC curve is used to describe the performance of the decision engine in discriminating between detection events $(\mathscr{s})$ and non-detection events $(\mathscr{n})$. The greater the area under the ROC curve, denoted $P(\mathscr{A})$, the better the performance of the decision engine. When using ROC curves, decision rules are generally made by varying a single control parameter, generating the ROC curve, and then comparing the curve with curves generated when using other values for the control parameter [76].

In cases where the prior probabilities $P(\mathcal{S})$ and $P(\mathcal{N})$ are unknown, (2.85) cannot be solved explicitly. In this case, the decision rule typically seeks to minimize the maximal conditional risks. This is called the minimax rule [76] [112]. A conditional risk is the cost of choosing a hypothesis when another hypothesis is true multiplied by the probability choosing that hypothesis when another is true.

There are situations where neither a priori probabilities (as with minimax hypothesis testing) nor realistic costs cannot be determined, or it would be desirable to impose costs [112]. To bypass this difficulty, conditional probabilities are used. These conditional probabilities include the probability of correct detection (a hit), $P_D = P(\mathcal{S}|\mathscr{s})$, and the probability of false alarm, $P_{FA} = P(\mathcal{S}|\mathscr{n})$. When these conditional probabilities are used, the goal of the decision engine is typically to constrain P_{FA} while maximizing P_D. This decision making approach is called the Neyman-Pearson criterion [76], [111]. Alternately, since the probability of a miss is equal to $1 - P_D$, the probability of a miss can be minimized instead of dealing explicitly with P_D. The Neyman-Pearson criterion is expressed as follows [112]:

$$\max_{\mathscr{B}} \left(P_D\left(\mathscr{B}\right)\right) \text{ subject to } P_{FA}\left(\mathscr{B}\right) \leq \alpha \tag{2.87}$$

where α is the constraint on the probability of false alarm. Several probability distributions, including the Gaussian distribution, can be used in obtaining analytic Neyman-Pearson decision criterion [76].

The discussion above only addressed simple hypothesis testing problems (only a single distribution for the observation, or evidence, is used), but in many hypothesis testing problems multiple distributions can occur under each of the hypotheses. This scenario is referred to as composite hypotheses [112].

Literature stresses the importance of selecting the best parameters to use in making detection or non-detection, or classification, decisions [76]. Sensors such as radar that decide between detection and non-detection may simply make use of the amplitude of the radar return in autonomous decisions that a detection occurred. However, a lot of detection problems may be treated as classification problems, where multiple parameters are available to choose from when trying to make autonomous detection decisions. Furthermore, there may be a need to make use of multiple parameters in order to obtain a "feature space" that can be partitioned into regions with boundaries that separate decision outputs. Discussion of a feature space is really a restating of the use of a multi-dimensional evidence variable $\vec{\chi}$. It should be pointed out that [76] states explicitly that hypothesis testing is not the same thing as pattern classification. Therefore, care is required in making use of pattern classification theory and terminology.

In the absense of analytically determined decision rules, a decision engine is trained using data where the truth is known [76]. As stated in [76], the most effective means by which to develop classifiers is to learn from example patterns through training.

2.8.2 Applicability of Signal Detection Theory to a Signal Decomposition Stopping Criteria. In the case of search grid decomposition that obtains estimates of multiple rays simultaneously using an iterative approach (referred to here as "multi-ray"), an ensemble of estimate ray waveforms is summed to yield an aggregate estimate of the GNSS direct path and multipath signals, thus an estimate search grid. In instances where C/N_0 is high, the estimate for the first ray ($m = 0$) will almost inevitably originate from the search grid direct path and/or multipath content. There are three possible outcomes as ray waveform estimate parameters are added to the ensemble: 1) iteration progresses the algorithm towards reaching the error minimum between the truth and the estimate, 2) iteration has progressed to the point where the error minimum is reached with the current iteration, or 3) the error minimum has already been reached and further iteration is not necesssary. Correct detection occurs in the first and second cases, and a false alarm occurs in the third case. In the first case, it is desired that the algorithm continue processing. In the second and third case, it is desired that the algorithm halts. To decide whether to continue decomposition iterations, an evidence variable must be determined. The evidence variable is denoted F_{p_ε}, and is discussed in Section 3.3.13. The state denoting a correct detection is denoted $\mathscr{s}$. The false alarm state is denoted $\mathscr{n}$.

The cost of a poor decision may very greatly, depending on the difference in error between the ray waveform ensemble decided to be the lowest in error, versus the ensemble that is truly lowest in error. If there is little difference in error, there may be little cost in making an incorrect decision.

2.8.3 Estimating the Number of Multipath Rays.

Fundamentally, the need for development of a stopping criteria is a product of the uncertainty regarding the number of multipath rays to estimate. This problem is discussed in literature on MEDLL [114] and SAGE [57], and presents itself in the case of SDPA as well.

In the general case of wireless communications signaling, literature outlines various techniques for estimation of the number of multipath rays present in a received signal. A method for estimation of the number of multipath rays as part of application of autoregressive multipath parameter estimation techniques is outlined in [115]. Another method for the estimation of number of multipath rays as part of a subspace-based method for estimating signal parameters is outlined in [116]. A similar method involves making use of eigenvalues from the covariance matrix of channel estimates [117] [118]. Still another method involves the determination of the number of multipath rays as a product of the use of the MUSIC algorithm and multi-aperture processing [119]. It is important to point out that, in general wireless communications signaling, the intent of the receiver is not to recover precise signal reception timing information, as with GNSS receiver signal processing, but to recover the original information output by the transmitter. Therefore, multipath estimation and mitigation techniques are couched in terms of the impact of multipath on bit values, rather than on ray waveform reception timing information. The signal models are the same between wireless communication and GNSS, but the intent of channel modeling research is different between the two fields.

In the case of GNSS receiver signal processing, [120] asserts that the Minimum Description length and the Generalized Likelihood Ratio Test are the most widely used means by which the number of mulitipath rays is estimated.

2.8.4 Signal Decomposition Stopping Criteria. An alternate approach to the problem of estimating the number of rays present in the received signal is to decompose the signal until a stopping criteria is reached. This approach, which is used in the research presented in this dissertation, involves estimating ray parameters until the residual remaining after comparing the received and estimate waveforms is reduced until a stopping criteria threshold is crossed.

Generally speaking, the application of a stopping criteria to a decomposition algorithm is considered to be ad hoc in nature. There are many examples in literature where a stopping criteria is applied to a signal decomposition algorithm. [121] outlines the consideration of a stopping criteria when using the Hilbert-Huang transform, which uses Empirical Mode Decomposition (EMD). The stopping criteria is considered extensively when using EMD [122] [123] [124]. In the case of the research reported in [125], EMD requires use of a stopping criteria that is determined through simulations and experiments. Ultrasonic Chirplet Signal Decomposition makes use of a stopping criteria based on the value of a signal reconstruction error or when a predefined number of iterations occurs [126]. The application of a stopping criterion as part of the application of the Perceptual Matching Pursuit algorithm for audio coding based on the amount of energy present in a residual is outlined in [127]. The algorithm itself is used in the parameterization and compression of audio signals. A similar approach for Matching Pursuit Decomposition (MPD) as it applies to the evaluation of life estimates of aerospace structures is considered in [128]. The use of MPD as it applies to parameterizing electric signals in power systems, to include the specification of a stopping criteria similarly to the other matching pursuit papers mentioned is outlined in [129]. Signal decomposition making use of Singular Value Decomposition (SVD) and requiring use of a stopping criterion is outlined in [130]. The use of a pre-existing number of iterations to be performed in decomposition, or the reduction in energy within a residual remaining after comparison of received and estimate signals is a recurring theme in stopping criteria literature.

2.9 GNSS Signal Simulators

Thombre et al. present a large, though not exhaustive, list of the various software-based GNSS signal simulators that have been constructed both in academia and in industry [131]. He offers a comparison and a contrast between various simulators, and provides a Simulink-based GNSS simulator design as well. As part of this Simulink-based simulator, a channel model is presented that would include five multipath components, as well as additive white Gaussian noise (AWGN). This channel model would make use of a channel impulse response function, seen in [71] for example, to generate multipath signals that are added to the simulated LOS signal. Thombre categorizes software-based GNSS simulators into three different generations, and states that that simulation of multipath effects is available in generally in the second and third generations.

According to [131], the first generation of software-based GNSS signal simulators did not typically make use of error modeling.

Three examples of what Thombre classifies as falling in the second generation of GNSS simulators are outlined in [132] [133] and [134]. Reference [132] describes simulation of multipath by generating up to seven rays for each transmitted GNSS signal in the simulation. Each ray is a variation of the LOS signal, with varied amplitude, phase, and propagation delay. Reference [133] describes simulation of two types of multipath, diffuse and specular. Diffuse multipath is said to result from near echoes (scattering), whereas specular multipath is said to result from far echoes (reflections). In 2.4.1.2, the term "fading" is used to describe what [133] calls diffuse multipath, and a Rayleigh distribution is used to model the amplitude of near echoes. Additionally, [133] uses ground reflection (the reflection of a GNSS transmission off the ground and then to the receive antenna) as an example of specular reflection. Specular reflection is addressed in 2.4.1.4, and in [133] a ray model is used to simulate the placement of reflectors around the receive antenna as GNSS space vehicles are simulated as transmitting navigation signals overhead. Reference [134] simulates multipath in terms of range measurement errors, rather than by adding probabilistic elements to simulated GNSS signals. These techniques, as described by Thombre, are relatively simple compared to those implemented in third generation software-based simulators.

Three examples of what Thombre classifies as falling in the third generation of GNSS simulators are outlined in [71], [135], and [136]. The details for the simulator outlined in [71], which make use of the DLR measurement campaign, are discussed in Section 2.4.2.2. Reference [135] describes the application of specular or statistical models making use of Rice or Rayleigh distributions. Reference [136] describes the use of a ray shooting algorithm based on electromagnetic propagation theory. This ray shooting approach makes use of the physics-based models, rather than signal simulation that makes use of probabilistic models. This physics-based approach was described in Section 2.4. The local environment is simulated using these models.

2.10 Structures of GNSS Signals Other than GPS

There are three global radionavigation satellite constellations other than GPS that are currently implemented or soon to be implemented. These constellations include GLONASS, Compass, and Galileo, arguably in order of constellation maturity. For the civil, or open, signals transmitted by the satellites associated with each of the four systems, the carrier frequency range employed by all the constellations ranges from approximately 1561.098 to 1605.375 MHz (between Compass B1 [137], GPS L1 [4], Galileo E1 [138], and GLONASS G1 [139]). These constellations make use of signals that are encoded with direct sequence spread spectrum (DSSS) encoding vectors, as does GPS.

To provide background for discussion of signals from constellations other than GPS, background on GPS is again provided. With GPS signals, a carrier frequency is used to modulate a baseband signal consisting of a direct sequence spread spectrum code vector using 1023 chips that is alternated in sign through binary phase shift keying [4] to provide the means of modulating the signal in order to transmit navigation data. This signal structure is very similar to the signals employed by other constellations.

GLONASS uses frequency division multiplexing to distinguish between signals from different satellites [139]. Among all the GLONASS open signals, the carrier frequencies are centered at 1602 MHz. Given the use of frequency division multiplexing, individual channels are employed for different satellites. These channels are separated by 0.5625 MHz. There is, however, a DSSS encoding sequence embedded in GLONASS open signals. These encoding sequences are of a length equal to 511 chips. GLONASS uses binary phase shift keying in order to modulate the signal for navigation data transmission.

Compass signals are transmitted at a carrier frequency of 1561.098 MHz [137] and contain DSSS encoding sequences that are of length equal to 2046 chips [140].

Compass uses quadrature phase shift keying to modulate the signal for navigation data transmission.

Galileo signals are transmitted at a carrier frequency of 1575.42 MHz (same as GPS) [138] and contain DSSS encoding sequences that are of length equal to 4092 chips. Galileo uses phase shift keying to modulate the signal for navigation data transmission. Galileo additionally makes use of binary offset carrier signals in order to reduce the potential of interfering with and being interfered with by GPS signals. This binary offset carrier simply involves the use of an additional square sub-carrier modulation similar to the DSSS encoding sequence in order to ensure distinction from GPS signal.

This information provides the means by which to compare the receiver signal processing methodology required for each of the four constellations.

2.11 Summary

This chapter outlines the theory underpinning the research in this dissertation. Previous research provides insight into the problem of GNSS multipath ray waveform parameter estimation, and also reveals that the specific problem being considered in this research (the development of a decomposition algorithm designed specifically to obtain parameters from which models can be derived) has not yet been addressed.

III. Signal Decomposition and Parameterization Algorithm (SDPA)

3.1 Introduction

This chapter outlines the details of SDPA. This algorithm was developed in research for this dissertation to decompose a received GNSS signal in order to obtain parameters for received multipath rays. This decomposition process has been documented and previously published [141] [142], and serves as the framework by which iterative decomposition takes place. This framework provides the flexibility required to accommodate variations in how signal parameters are obtained. This flexibility allows for interchangeability of the SDPA search space decomposition methodology with the SAGE-based decomposition process described in Section 2.6.3. Because of this, side-by-side estimation performance comparisons can be fairly conducted. The results of these comparisons are discussed in Chapter IV.

An introduction to the algorithm that describes the top-level concept employed in this algorithm and the decomposition process underpinning the algorithm is provided in Section 3.2. The top-level flowchart illustrating the process that is implemented and traced to the top-level concept is provided in Section 3.3. This section presents the mechanics associated with the algorithm itself, with a canonical description of the operations within the algorithm. The reader of this dissertation should be able to reconstruct SDPA given this section of the dissertation. Finally, Section 3.4 provides a comparison of SDPA- and SAGE-based decomposition algorithms.

3.2 Algorithm Concept

Figure 3.1 illustrates a top-level flowchart of the intra-PIT signal decomposition algorithm. This figure corresponds directly with the process flowchart illustrated in Figure 3.2. The concept behind the algorithm is simple. First, RF front end functions (such as frequency downconversion and signal sampling) are performed. Then, the signal is aligned so that data is processed by the decomposition algorithm only in integer multiples of the duration of the PRN code sequence present in GPS L1 C/A-coded signals (so integer multiples of 1 msec). The decomposition algorithm is only to process whole periods of the PRN code sequence (so a PIT-sized data vector is processed, where the first sample in the data vector is from the first chip in the PRN sequence, and the last sample in the vector is from the last chip in the PRN sequence). Once alignment takes place, received data that is sized to be of PIT-sized duration is input for search grid computation and initial ray parameter estimation. The amplitude, the Doppler frequency, the relative propagation delay, and the carrier phase are then coarsely estimated. These coarse parameter estimates are then used to initiate fine estimation of these four parameters for a ray waveform using a stochastic search and optimization technique (simulated annealing), and the portion of the search grid associated with the estimated parameters is removed from the search grid. The remainder of the search grid is then evaluated to obtain a stopping criteria statistic. The stopping criteria statistic is obtained by comparing the peak power of the search grid with the noise power present in the search grid. The stopping criteria statistic is then evaluated to determine if the stopping criteria is satisfied. If the criteria is satisfied, this signifies that all rays present in the search grid are treated as having been estimated by the algorithm, and the algorithm advances to the next PIT for decomposition. If the criteria is not satisfied, the search grid is evaluated to compute the estimate error, and the algorithm processor decides if the estimate error can be further reduced. If so, decomposition of the current PIT continues, and the matrix difference between the received and estimate search grids is computed for use in subsequent coarse ray parameter estimation. This continues until there are no more rays to estimate. If error cannot be reduced, all rays present in the search grid are treated as having been estimated by the algorithm, and the algorithm advances to the next PIT for decomposition. This loop takes place until all data is processed and parameterized.

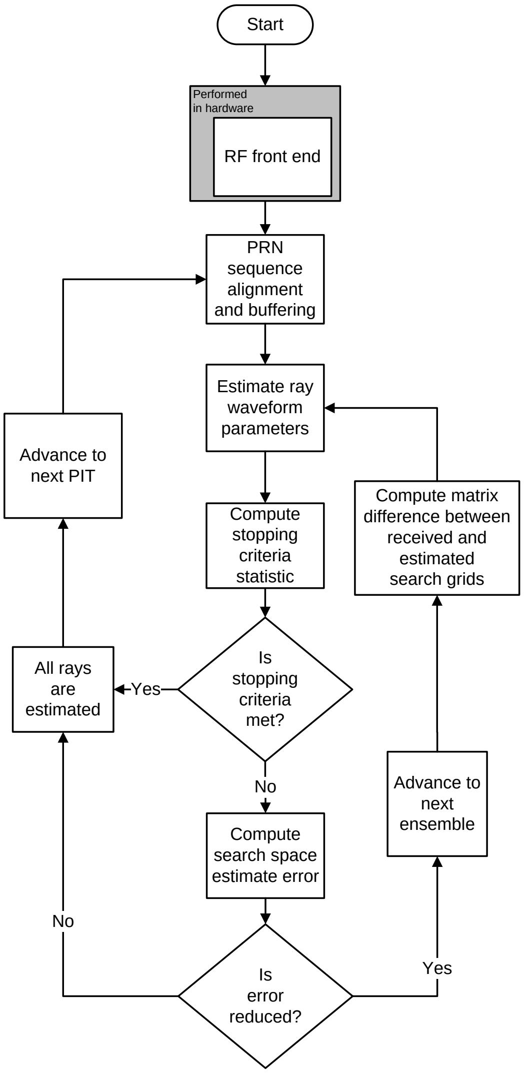
Start
Performed in hardware
RF front end
PRN sequence alignment and buffering
Estimate ray waveform parameters
Compute stopping criteria statistic
Is stopping criteria met?
Yes
All rays are estimated
Advance to next PIT
No
Compute search space estimate error
Is error reduced?
No
Yes
Advance to next ensemble
Compute matrix difference between received and estimated search grids

Figure 3.1: Top-level flowchart illustrating intra-PIT signal decomposition algorithm.

A previous version of SDPA made use of hard estimate decisions on ray waveform parameter estimates [141]. However, this methodology (referred to here as "SDPA single ray estimation") was determined to be inadequate in obtaining estimates of the received search space that explicitly present destructive multipath interference. A revision of the SDPA algorithm has been undertaken to make use of soft ray waveform parameter estimate decisions that are obtained simultaneously through the parallel estimation of multiple ray waveform parameters, as part of the stochastic search block in Figure 3.1. This approach to ray waveform parameter estimation is referred to as "multiray" estimation. This approach to waveform parameter estimation complicates the notation used to describe the algorithm, relative to the notation used in [141].

It is important to point out that the intent of SDPA is to replicate ray waveform parameters from received data with the intent of constructing a model that describes these parameters over time. This model could then be used in the construction of a simulator that provides a better assurance of the presence of what could be expected from a real-world environment than would would be the case using existing simulators. SDPA is not designed to obtain precise estimates for individual ray waveform parameters. Therefore, it is possible that ray waveform parameter estimates obtained using SDPA may not accurately reflect the ray waveform parameters that truly comprise received signals.

3.3 Process Flowchart

Figure 3.2 illustrates the flowchart outlining the process used to estimate the multipath-propagated rays received with a software receiver. This flowchart involves the execution of several processes, corresponding with sections of this chapter where discussion of the specifics of each of these processes is provided. The processes illustrated in the flowchart in Figure 3.2 generally correspond with the processes in the flowchart in Figure 3.1. The processes in Figure 3.2 are as follows: downconversion of received data to an intermediate frequency (Section 3.3.2), sampling of data (Section 3.3.3), buffering of code-aligned data (Section 3.3.4), computation of the search grid from received data (Section 3.3.5), estimation of the initial search grid peak location (Section 3.3.6), estimation of initial parameters (Section 3.3.7), decomposition of the search space (Section 3.3.8), computation of the estimate search space (Section 3.3.9), comparison of received and estimate search spaces (Section 3.3.10), computation of the estimate error (Section 3.3.11), output of parameters to the ray database (Section 3.3.12), computation of the stopping criteria statistic (Section 3.3.13), determination of alternate search grid comparison regional maxima to consider (Section 3.3.14), and initial estimation of search grid alternate peak locations (Section 3.3.15). There are a few other processes and decisions for which the flowchart process label is self-explanatory, such as "$p = p + 1$" and "$\varepsilon = \varepsilon + 1$", which are used simply to increment the data vector to consider (in terms of the PIT p, where the number of PIT-sized data vectors contained in the data set is $p_{\max}$) and the multipath ray parameter ensemble to consider (where ε is the index for the multipath ray parameter ensemble within PIT p), respectively. The decision points "Is $\varepsilon = 0$", "Is $E_{p_m} \leq E_{p_{m-1}}$", "Is $F_{p_\varepsilon} < \Upsilon$", and "Is $p > p_{\max}$" are each outlined within the section associated with the process that leads to the decision. The search space decomposition algorithm will be outlined with an additional flowchart as part of the section that addresses that process (Section 3.3.8).

As stated previously, the processes illustrated in the flowchart in Figure 3.2 generally correspond with the processes in the flowchart in Figure 3.1. The reason two flowcharts are provided is to separate the functional description from the architectural description of the algorithm. The functional description is provided in the flowchart in Figure 3.1 and the architectural description of the algorithm is provided in Figure 3.2. These two flowcharts, respectively, are to communicate a conceptual and a mechanical understanding of the algorithm. To understand the process flow as narrated in this chapter, the flowchart in Figure 3.2 will be of particular use. The correspondence between the two flowcharts can be traced as follows:

- The portions that are performed in hardware (and surrounded by shaded boxes in each of the flowcharts) correspond with each other.
- "PRN sequence alignment and buffering" in Figure 3.1 corresponds with "Code aligning buffer" in Figure 3.2.
- "Estimate ray waveform parameters" in Figure 3.1 corresponds with "Received data search grid computation," "Initial search grid peak location estimation," "Initial parameter estimation," and "Search space decomposition" in Figure 3.2.

- “Compute stopping criteria statistic” processes in each flowchart correspond with each other.

- “Is stopping criteria met?” in Figure 3.1 corresponds with “Is $F_{p_\varepsilon} < \Upsilon$” in Figure 3.2.

- “Compute search space estimate error” in Figure 3.1 corresponds with “Compute estimate search space,” “Compare received and estimate search grids” and “Compute estimate error” in Figure 3.2.

- “Is error reduced?” in Figure 3.1 corresponds with “Is $E_{p_\varepsilon} \leq E_{p_{\varepsilon-1}}$” in Figure 3.2.

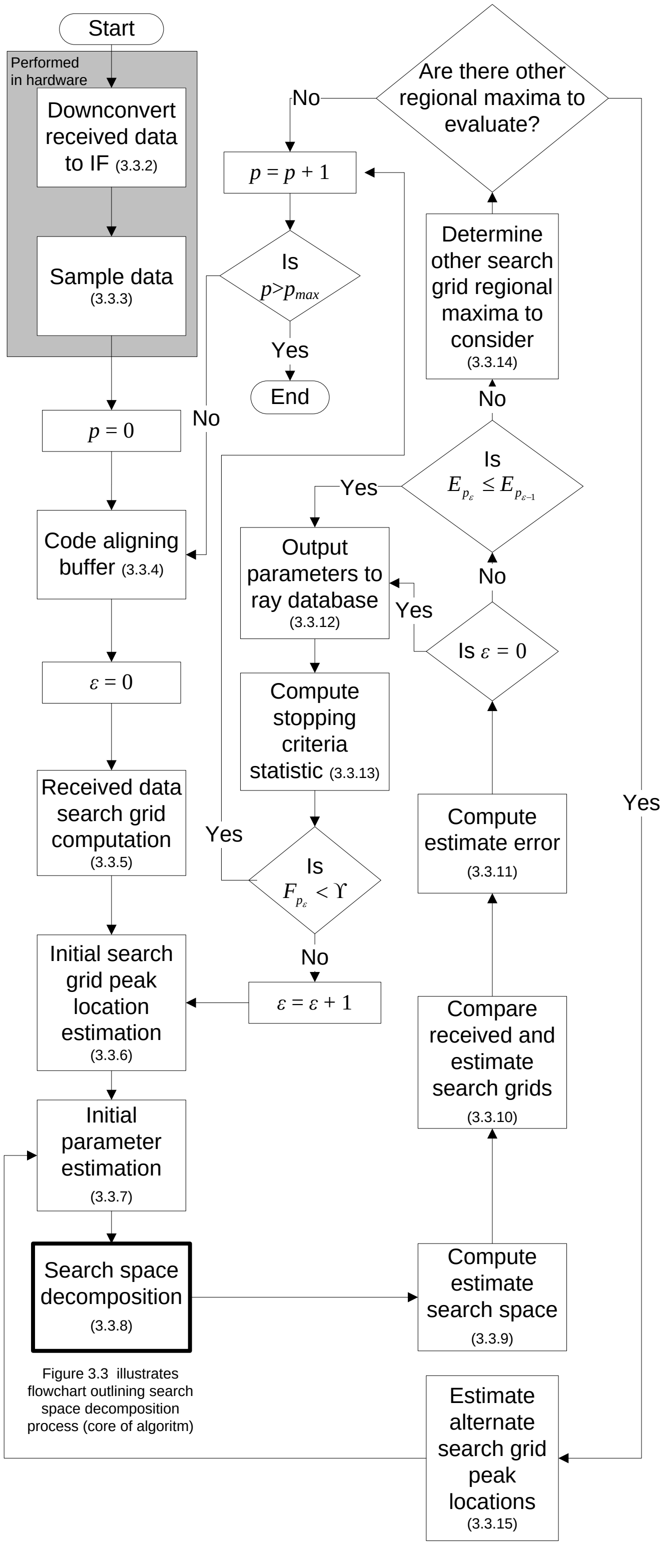
Start
Performed in hardware
Downconvert received data to IF (3.3.2)
Sample data (3.3.3)
$p = 0$
Code aligning buffer (3.3.4)
$\varepsilon = 0$
Received data search grid computation (3.3.5)
Initial search grid peak location estimation (3.3.6)
Initial parameter estimation (3.3.7)
Search space decomposition (3.3.8)
Figure 3.3 illustrates flowchart outlining search space decomposition process (core of algoritm)
$p = p + 1$
Is $p>p_{max}$
Yes
End
No
Output parameters to ray database (3.3.12)
Compute stopping criteria statistic (3.3.13)
Is $F_{p_\varepsilon} < \Upsilon$
Yes
No
$\varepsilon = \varepsilon + 1$
Are there other regional maxima to evaluate?
No
Yes
Determine other search grid regional maxima to consider (3.3.14)
No
Is $E_{p_\varepsilon} \le E_{p_{\varepsilon-1}}$
Yes
No
Is $\varepsilon = 0$
Yes
Compute estimate error (3.3.11)
Compare received and estimate search grids (3.3.10)
Compute estimate search space (3.3.9)
Estimate alternate search grid peak locations (3.3.15)

Figure 3.2: Flowchart illustrating the signal parameterization algorithm. Processes are outlined in Section 3.3, with exceptions in cases where the flowchart process label is self-explanatory, such as "$p = p + 1$" and "$\varepsilon = \varepsilon + 1$". The section number for the discussion corresponding with each block is provided in the flowchart.

- “Advance to next ensemble” in Figure 3.1 corresponds with “$\varepsilon = \varepsilon + 1$” in Figure 3.2.
- “Compute matrix difference between received and estimate search grids” in Figure 3.1 corresponds with “Compute estimate search space” and with subprocesses within “Initial search grid peak location estimation” in Figure 3.2.
- “All rays are estimated” in Figure 3.1 corresponds with “Output parameters to ray database” in Figure 3.2.
- “Advance to next PIT” in Figure 3.1 corresponds with “$p = p + 1$” in Figure 3.2.

3.3.1 Ray Model of Received Signal. In this algorithm description, the following model will be used for the continuous time-domain received signal $r(t)$, following the development for (2.10) in Section 2.2.3:

$$r(t) = \sum_{m=0}^{M(t)} A_m(t) D(t - \tau_m) x(t - \tau_m) \cos\left(2\pi\left(f_{L1} + f_{D_m}(t)\right) t + \phi_{r_m}\right) + \eta(t) \qquad (3.1)$$

where m is the index of the multipath ray and the LOS ray is the 0th multipath ray (so $m = 0$ for the ray assumed to be LOS), $M(t)$ is the number of multipath rays at time t, $A_m(t)$ is the peak amplitude of the signal from the mth ray at time t, $D(t - \tau_m)$ is the value of the navigation data message symbol at time $t - \tau_m$, τ_m is the propagation delay of the mth ray between the transmitting SV and the receiver, $x(t-\tau_m)$ is the value of the C/A-coded PRN sequence chip value at time $t-\tau_m$, f_{L1} is the center frequency of the GPS L1 band, $f_{D_m}(t)$ is the Doppler frequency offset from the L1 center frequency at time t, ϕ_{r_m} is the carrier phase of the mth multipath ray, and $\eta(t)$ is the white Gaussian noise that is added to the received LOS and multipath signals at time t. This model will be used to fundamentally describe the signal at entry into the receiver throughout the remainder of this section.

3.3.2 Downconverting Received Data to an Intermediate Frequency. This process is executed in the RF front end hardware of the software receiver. This section is provided to present the model that is used to describe the output of the downconversion process. As described in 2.3.1.1, this process accepts $r(t)$ as an input and outputs the downconverted received signal $r_{\mathrm{IF}}(t)$ whose band has a center frequency of f_{IF}. There are frequencies other than f_{L1} that are available for various GPS users, but for the purposes of this description of the multipath estimation algorithm, only L1 will be considered. The equation used to describe the output of the downconversion process is as follows:

$$r_{\mathrm{IF}}(t) = \sum_{m=0}^{M(t)} A_m(t) D(t-\tau_m) x(t-\tau_m) \cos\left(2\pi\left(f_{\mathrm{IF}} + f_{D_m}(t)\right)t + \phi_m\right) + \eta(t) \tag{3.2}$$

where f_{IF} is the intermediate frequency of the receiver, and ϕ_m is the carrier phase of the mth multipath ray at the input into the ADC. It is assumed that $r_{\mathrm{IF}}(t)$ is filtered using an ideal bandpass filter centered at f_{IF}. GNSS receivers such as the TRIGR typically use an intermediate frequency when outputting downconverted and sampled signals for processing by the user. This process within a software receiver is transparent to the user. Note at this point that the carrier phase of the multipath ray input to the ADC may be completely different from the carrier phase of the original received signal, thus the distinction between ϕ_{r_m} in (3.1) and ϕ_m in (3.2). This does not impact how the signal is processed. Again, the equation above is a model that makes use of the assumption of ideal bandpass filtering in its development.

This equation for $r(t)$ sufficiently serves as the model for the TRIGR receiver, which processes real-valued data from received signals, but does not describe the model to be used for a receiver that makes use of complex-valued data. A receiver that makes use of complex-valued data processes a simultaneous in-phase signal $i(t)$ and quadrature signal $q(t)$. The model to describe continuous time-domain data where in-phase and quadrature signals are both present is as follows:

$$r_{\text{IF}}(t) = i(t) + jq(t) + \eta_i(t) + j\eta_q(t) \tag{3.3}$$

where $j = \sqrt{-1}$. Note the presence of complex-valued noise in this model. This is done to account for the presence of zero-mean white Gaussian noise that is added independently to both the in-phase and quadrature signals. The noise on the in-phase signal and the noise on the quadrature signal are independent of each other, and share the same variance (referred to as "circular"). The continuous time-domain in-phase and quadrature signals $i(t)$ and $q(t)$ are modeled as follows:

$$i(t) = \sum_{m=0}^{M(t)} A_m(t)D(t-\tau_m)x(t-\tau_m)\cos\left(2\pi\left(f_{\text{IF}} + f_{D_m}(t)\right)t + \phi_m\right) \tag{3.4}$$

and

$$q(t) = \sum_{m=0}^{M(t)} A_m(t)D(t-\tau_m)x(t-\tau_m)\sin\left(2\pi\left(f_{\text{IF}} + f_{D_m}(t)\right)t + \phi_m\right) \tag{3.5}$$

$r_{\text{IF}}(t) \in \mathbb{C}$ is now restated using a complex exponential instead of sine and cosine functions:

$$r_{\text{IF}}(t) = \sum_{m=0}^{M(t)} A_m(t)D(t-\tau_m)x(t-\tau_m)\exp\left(j\left(2\pi\left(f_{\text{IF}} + f_{D_m}(t)\right)t + \phi_m\right)\right) + \eta_{iq}(t) \tag{3.6}$$

where

$$\eta_{iq}(t) = \eta_i(t) + j\eta_q(t) \tag{3.7}$$

$r_{\text{IF}}(t) \in \mathbb{C}$ is further restated by dividing the complex exponential term into two separate terms for the carrier frequency and the carrier phase as follows:

$$r_{\text{IF}}(t) = \sum_{m=0}^{M(t)} A_m(t)D(t-\tau_m)x(t-\tau_m)\exp\left(j\left(2\pi\left(f_{\text{IF}} + f_{D_m}(t)\right)t\right)\right)\exp\left(j\phi_m\right) + \eta_{iq}(t) \tag{3.8}$$

3.3.3 Sampling Data. This process is executed in the front end hardware of the software receiver. This section is provided to present the model that is used to describe the output of the sampling process. As described in Section 2.3.1.2, this process accepts $r_{\mathrm{IF}}(t)$ as an input and outputs the discrete time-domain received signal $r[n]$. For bookkeeping purposes, the PIT index value will be initialized by asserting that $p = 0$. The equation used to express the sampling of $r_{\mathrm{IF}}(t)$, which yields $r[n]$, is as follows.

$$r[n] = r_{\mathrm{IF}}(nT_s), n \in \mathbb{N} \tag{3.9}$$

where n is the sample index and T_s is the sampling period. This process within a software receiver is transparent to the user. This yields the following result:

$$r[n] = \left(\sum_{m=0}^{M[n]} A_m[n] D_m[n] x_m[n] f_m[n] \theta_m[n] \right) + \eta[n] \tag{3.10}$$

$$M[n] = M(nT_s) \tag{3.11}$$

$$A_m[n] = A_m(nT_s) \tag{3.12}$$

$$D_m[n] = D(nT_s - \tau_m) \tag{3.13}$$

$$x_m[n] = x(nT_s - \tau_m) \tag{3.14}$$

$$f_m[n] = \exp\left(j\left(2\pi\left(f_{\mathrm{IF}} + f_{D_m}(nT_s)\right) nT_s\right)\right) \tag{3.15}$$

$$\theta_m[n] = \exp\left(j\phi_m\right) \tag{3.16}$$

$$\eta[n] = \eta_{iq}(nT_s) \tag{3.17}$$

Note the use of notation where only the discrete-time index is used (as in $M[n]$), rather than the use of the sampling period multiplied by the discrete-time index (as in $M(nT_s)$). This is to make use of the notation convention used in [7] to indicate that these equations, as well as subsequent equations that make use of the discrete-time index alone in brackets, compose vectors of values over discrete time instances.

3.3.4 Buffering of Code-aligned Data. This process accepts $r[n]$ and p as inputs and outputs the buffered vector of received data $\mathbf{r}_p$ of size N_p. The definition of code alignment is explained further in this section, particularly to describe (3.29). As part of this process, for bookkeeping purposes, the ensemble index value will be initialized by asserting that $\varepsilon = 0$.

The code-aligning buffer processes the discrete time-domain received data $r[n]$, where $r[n] \in \{\mathbb{R}, \mathbb{C}\}$, and divides $r[n]$ into vectors of size N_p. Therefore, $\mathbf{r}_p$ is expressed as follows:

$$\mathbf{r}_p = [r[n]]_{n=pN_p}^{N_p-1+pN_p}, p \in \mathbb{N} \tag{3.18}$$

However, this can be restated as:

$$\mathbf{r}_p = \left[\sum_{m=0}^{M[n]} A_m[n]D_m[n]x_m[n]f_m[n]\theta_m[n] + \eta[n]\right]_{n=pN_p}^{N_p-1+pN_p} \tag{3.19}$$

Because of this, (3.18) can be restated as a sum of individual vectors as follows:

$$\mathbf{r}_p = \left(\sum_{m=0}^{M_p} \mathbf{r}_{p_m}\right) + \boldsymbol{\eta}_p \tag{3.20}$$

where

$$\mathbf{r}_{p_m} = [A_m[n]D_m[n]x_m[n]f_m[n]\theta_m[n]]_{n=pN_p}^{N_p-1+pN_p} \tag{3.21}$$

and

$$\boldsymbol{\eta}_p = [\eta[n]]_{n=pN_p}^{N_p-1+pN_p} \tag{3.22}$$

Note this expression is constrained by making the assumption that the number of rays is fixed within one PIT, thus $M_p = M[n] \forall n \in \{pN_p, pN_p+1, \ldots, N_p-1+pN_p\}$. (3.21) is restated as follows:

$$\mathbf{r}_{p_m} = A_{p_m}D_{p_m}\mathbf{x}_{p_m} \circ \mathbf{f}_{p_m}\theta_{p_m} \tag{3.23}$$

where $\circ$ denotes the Hadamard product (element-by-element array multiplication). The amplitude A_{p_m} is constrained by assuming that the amplitude for each ray is fixed within one PIT. The same constraining assumption is made for the navigation data message bit value D_{p_m} and the complex exponential carrier phase term θ_{p_m}.

$$A_{p_m} = A_m[n] \forall n \in \{pN_p, pN_p + 1, \ldots, N_p - 1 + pN_p\} \tag{3.24}$$

$$D_{p_m} = D_m[n] \forall n \in \{pN_p, pN_p + 1, \ldots, N_p - 1 + pN_p\} \tag{3.25}$$

$$\mathbf{x}_{p_m} = [x_m[n]]_{n=pN_p}^{N_p-1+pN_p} \tag{3.26}$$

$$\mathbf{f}_{p_m} = [f_m[n]]_{n=pN_p}^{N_p-1+pN_p} \tag{3.27}$$

$$\theta_{p_m} = \theta_m[n] \forall n \in \{pN_p, pN_p + 1, \ldots, N_p - 1 + pN_p\} \tag{3.28}$$

Given the C/A-coded PRN sequence vector $\mathbf{b}$ (of size $N_c = 1023$) that is modulated within the GPS waveform to spread the signal spectrum, alignment of the C/A-coded PRN sequence from one PIT p to the next will be forced. This is done by simply assuming that the most prominent ray received in $\mathbf{r}_{p_m}$ (which is assumed to be the LOS ray denoted as $m = 0$) is output from the code-aligning buffer with the following constraint in place:

$$\mathbf{x}_{p_0} = [b_0, \ldots, b_{N_c-1}] \ \forall p \tag{3.29}$$

This means that the first element of the coding vector $\mathbf{x}_{p_m}$ for the LOS ray would contain the value of the first chip in the C/A-coded PRN sequence vector, and the last element in $\mathbf{x}_{p_m}$ would contain the value of the last chip in the sequence vector. This assumption is not generally going to be the case in the practical processing of received signals, but is assumed for the purpose of this description.

This aligning function can be performed by simply processing the received GPS data using a data surveying utility (such as a simple GPS software receiver tracking loop), and then tracking points in the signal where $x[n] = b_0$, while $x[n-1] = b_{N_c-1}$.

This function is performed the same way regardless of the value of PIT p.

3.3.5 Computation of the Search Grid from Received Data. This process accepts $\mathbf{r}_p$ and N_p as inputs and outputs the search space $\mathbf{R}_p$ for the received signal vector $\mathbf{r}_p$. The search space may also be referred to here as the search grid. Because the same algorithm will be used for computing the search grid in several instances, with variation potentially in the propagation delay (or alternately path length offset) window size τ_{window} (or alternately δd_{window}), the frequency window size f_{window}, the propagation delay grid spacing (or alternately path length offset grid spacing) $\Delta\tau$ (or alternately $\Delta(\delta d)$) and the frequency grid spacing Δf, the "grid" operator will be defined to describe the search space operation, which yields an arbitrary search grid $\mathbf{S}$:

$$\mathbf{S} = \text{grid}(\mathbf{s}, \tau_{window}, f_{window}, \Delta\tau_{input}, \Delta f_{input}) \tag{3.30}$$

where $\mathbf{s}$ is a vector of PIT size that is input into the search space computation operation. The input value Δf_{input} is the desired frequency grid spacing, as defined by the user. The input value $\Delta\tau_{input}$ is the desired propagation delay grid spacing, as defined by the user. The actual grid spacing of the search space, in both propagation delay and frequency, will be no greater than the values input by the user for grid spacing.

Before explaining how the search grid is computed for the process outlined in this section of the document, it is asserted that the search grid $\mathbf{R}_p$ for received PIT-sized vector $\mathbf{r}_p$ is computed as follows:

$$\mathbf{R}_p = \text{grid}(\mathbf{r}_p, \tau_{window}, f_{window}, \Delta\tau_{input}, \Delta f_{input}) \tag{3.31}$$

3.3.5.1 Generalized Computation of the Search Space. This section outlines how the "grid" operator is used to compute the search space before making use of the operator to describe how the search space is computed. The search space is described in GNSS literature [1] [143], though variations on the search grid may be referred to in literature as the "ambiguity function" or "spectrogram." The equations defining how search space is constructed for this research, as presented in (3.32) - (3.47), are original work and have not been uncovered explicitly in literature. This methodology for the search space design technique is inspired by discussion on the subject found in [144].

The propagation delay window size τ_{window} is a user-dictated design choice. The choice of $\tau_{window} = 2T_c$ is made for this process in order to obtain estimates of the multipath rays adjacent to the LOS ray, in order to characterize the channel for the local environment around the receiver. As described in Section 2.4.2, multipath much more prominently affects receiver processing when a multipath ray is received with a propagation delay of less than T_c, relative to the LOS ray. Since the delay between the peak of a ray in the search space and the point where the ray is negligible in power is T_c, the sum of $T_c + T_c$ is used for the window size. However, this process can easily be executed for any arbitrary choice of propagation delay without requiring any changes to the algorithm. The conversion between τ_{window} and the path length offset window size δd_{window} is as follows:

$$\delta d_{window} = c\tau_{window} \tag{3.32}$$

where c is the speed of light.

The frequency window size f_{window} is a user-dictated design choice. The choice of $f_{window} = 10$ kHz is made because literature states [67] that Doppler shifts of up to $f_{window} = 10$ kHz can be observed in received signals.

The propagation delay grid spacing $\Delta\tau_{input}$ is a user-dictated design choice. The choice of $\Delta\tau_{input} = T_s$ is made because this value of $\Delta\tau$ corresponds with the actual resolution afforded by the sample rate of the receiver. The conversion between $\Delta\tau = T_s$ and the path length offset grid spacing $\Delta(\delta d)$ is as follows:

$$\Delta(\delta d) = c\Delta\tau \tag{3.33}$$

The propagation delay grid spacing $\Delta\tau$ that is actually used in computation of the search grid is to always be no larger than the grid spacing specified by the user. The equation for calculation of $\Delta\tau$ is as follows:

$$\Delta\tau = \frac{2\tau_{window}}{N_\tau - 1} \tag{3.34}$$

where

$$N_\tau = 2\left\lceil \max\left(f_s\tau_{window}, \frac{\tau_{window}}{\Delta\tau_{input}}\right)\right\rceil + 1 \tag{3.35}$$

The frequency grid spacing Δf that is actually used in computation of the search grid is to always exceed the grid spacing dictated by the user. The smaller the grid spacing, the better the initial estimate of the Doppler frequency offset that can be made for each ray in the PIT, thus the grid spacing choice of $\Delta f_{input} = 10$ Hz. The search grid can be computed several ways, generally making use of a fast Fourier transform (FFT). Therefore, the potential use of the FFT will be considered when computing Δf, but will not be considered when presenting the general form for the search grid in (3.41). The equation for calculation of Δf is as follows:

$$\Delta f = \frac{f_s}{N_{fft}} \tag{3.36}$$

where

$$N_{fft} = \max\left(N_s, 2^{\left(\left\lceil \log_2\left(\frac{f_s}{\Delta f_{input}}\right)\right\rceil\right)}\right) \tag{3.37}$$

where N_s is the size of the data vector $\mathbf{s}$ input to the "grid" operation and $\lceil\bullet\rceil$ indicates a ceiling rounding operation (rounding up to the nearest integer on the real number line more than the value contained in the modified brackets). Note that the function expressed with this equation is designed to make use of the computational efficiency afforded the FFT function when input vectors to an FFT function are of a length equal to a power of two.

The fast Fourier transform algorithm "fft" will not be further described in this prospectus, other than to say that the following operator is used to denote the FFT operation, where $\mathbf{s}_f$ is the vector output of the FFT operation:

$$\mathbf{s}_f = \text{fft}(\mathbf{s}, N_{fft})_{f \in [f_{low}, f_{high}]} \tag{3.38}$$

where N_{fft} is the number of points in the input vector in the FFT, and is as expressed in (3.37). The expression $f \in [f_{low}, f_{high}]$ denotes the range of frequencies, between f_{low} and f_{high}, for which data will be output in $\mathbf{s}_f$. Put another way, the vector output from an FFT operation is associated with a range of frequencies as follows: $f \in [-f_s/2, f_s/2)$. The subscript to the fft operator is used to denote that the range of frequency-domain points output in $\mathbf{s}_f$ is limited to $[f_{low}, f_{high}]$, rather than the entire range that would be output were the FFT not constrained. The number of points in the FFT whose frequencies are in the range $f \in [f_{low}, f_{high}]$ is denoted N_f. The following vector $\boldsymbol{\gamma}$ is defined to be the frequency values for each column in the search space:

$$\boldsymbol{\gamma} = [\gamma_z]_{z=0}^{N_f-1}, \gamma_z \in [f_{low}, f_{high}] \tag{3.39}$$

The code delay vector $\mathbf{x}_s$ for code delay τ_s is defined as follows:

$$\mathbf{x}_s(\tau_s) = \left[b \left[\left\lfloor N_c \left(\frac{wT_s - \tau_s}{N_c T_c} - \left\lfloor \frac{wT_s - \tau_s}{N_c T_c} \right\rfloor \right) \right\rfloor \right] \right]_{w=0}^{N_s-1}, b \in \{-1, +1\} \tag{3.40}$$

where τ_s is an arbitrary propagation delay input into the expression, $b[\bullet]$ is an element of the PRN sequence vector $\mathbf{b}$ designated by the natural number inside the brackets,$\lfloor \bullet \rfloor$ indicates a floor rounding operation (rounding down to the nearest integer on the real number line less than the value contained in the modified brackets) ,N_c is the number of chips in the PRN sequence, T_c is the chip duration (so $T_c = 0.001/N_c = 0.001/1023$ seconds for GPS C/A code), and N_s is as used in (3.37).

The search space computation operator "grid" and the output of the "grid" operator, $\mathbf{S}$, is defined as follows, where the "grid" operator for $\mathbf{S}$ is first presented in

(3.30) to generally relate the "grid" operator to (3.31):

$$\begin{aligned}\mathbf{S} &= \text{grid}(\mathbf{s}, \tau_{window}, f_{window}, \Delta\tau_{input}, \Delta f_{input}) \\ &= \begin{bmatrix} S(\beta_0,\gamma_0) & S(\beta_0,\gamma_1) & \cdots & S(\beta_0,\gamma_{N_f-1}) \\ S(\beta_1,\gamma_0) & S(\beta_1,\gamma_1) & \cdots & S(\beta_1,\gamma_{N_f-1}) \\ \vdots & \vdots & \ddots & \vdots \\ S(\beta_{N_\tau-1},\gamma_0) & S(\beta_{N_\tau-1},\gamma_1) & \cdots & S(\beta_{N_\tau-1},\gamma_{N_f-1}) \end{bmatrix}_{N_\tau \times N_f}\end{aligned} \tag{3.41}$$

where N_τ is the size of the vector of code delays considered in $\mathbf{S}$, thus the number of rows in $\mathbf{S}$, and is equal to the size of the following vector $\boldsymbol{\beta}$ defining the code delay values for each row in the search space:

$$\boldsymbol{\beta} = [-\tau_{window}, -\tau_{window} + \Delta\tau, -\tau_{window} + 2\Delta\tau, \ldots, \tau_{window}]^T \tag{3.42}$$

$$= \left[[\beta_y]_{y=0}^{N_\tau-1} \right]^T \tag{3.43}$$

To compute $S(\beta_y, \gamma_z)$, the following equation is used:

$$S(\beta_y,\gamma_z) = \begin{cases} S_i(\beta_y,\gamma_z) + jS_q(\beta_y,\gamma_z), & \mathbf{s} \in \mathbb{C} \\ 2\left(S_i(\beta_y,\gamma_z) + jS_q(\beta_y,\gamma_z)\right), & \mathbf{s} \in \mathbb{R} \end{cases} \tag{3.44}$$

where

$$S_i(\beta_y,\gamma_z) = \frac{1}{N_s}\sum_{w=0}^{N_s-1}\left(\left(\text{Re}(s_w)\cos(\phi_{s_w}(\gamma_y)) + \text{Im}(s_w)\sin(\phi_{s_w}(\gamma_y))\right)x_{s_w}(\beta_y)\right) \tag{3.45}$$

$$S_q(\beta_y,\gamma_z) = \frac{1}{N_s}\sum_{w=0}^{N_s-1}\left(\left(\text{Im}(s_w)\cos(\phi_{s_w}(\gamma_y)) - \text{Re}(s_w)\sin(\phi_{s_w}(\gamma_y))\right)x_{s_w}(\beta_y)\right) \tag{3.46}$$

and

$$\phi_{s_w}(\gamma_y) = 2\pi(f_{\text{IF}} + \gamma_y)T_s w \tag{3.47}$$

The multiplication by two of individual elements in $\mathbf{S}$ when $\mathbf{s} \in \mathbb{R}$ in (3.44) is required to obtain proper amplitude estimates for individual ray waveforms in $\mathbf{S}$, as signal power is halved when only the real-valued portion of $\mathbf{r}_p$ is decomposed, versus when $\mathbf{s} \in \mathbb{C}$.

3.3.6 Estimation of the Initial Search Grid Peak Location. In the case where the ensemble index $\varepsilon = 0$, this process accepts $\mathbf{R}_p$ as an input and outputs the comparison matrix $\boldsymbol{\chi}_{p_\varepsilon}$ and the row and column indices corresponding to the element in $\boldsymbol{\chi}_{p_\varepsilon}$ that has the highest magnitude. In cases where $\varepsilon > 0$, the estimate search grid output from the prior search space decomposition $(\varepsilon - 1)$, $\hat{\mathbf{R}}_{p_{\varepsilon-1}}$, is additionally input. $\hat{\mathbf{R}}_{p_\varepsilon}$ is the search grid computed from the summation of the discrete time-domain vectors associated with all previously obtained ray estimates. $\hat{\mathbf{R}}_{p_\varepsilon}$ contains aggregated search grid information for ε previously obtained ensembles.

In Section 3.3.5, the number of rows in the search space matrix $\mathbf{R}_p$ is established to be N_τ and the number of columns in $\mathbf{R}_p$ to be N_f. $\hat{\mathbf{R}}_{p_{\varepsilon-1}}$ is additionally asserted in this process to be of the same size, and that the frequency and code phase values of element positions shared between the two matrices are the same. The comparison matrix $\boldsymbol{\chi}_{p_\varepsilon}$ is now defined as follows:

$$\boldsymbol{\chi}_{p_\varepsilon} = \begin{cases} \mathbf{R}_p - \hat{\mathbf{R}}_{p_{\varepsilon-1}}, & \varepsilon > 0 \\ \mathbf{R}_p, & \varepsilon = 0 \end{cases} \tag{3.48}$$

Defining $\boldsymbol{\chi}_{p_\varepsilon}$ as shown in (3.48) simply provides a means by which the received search space is compared with the most recently computed search space of the sum of estimated multipath rays. As ε increases, the average magnitude of each element of $\boldsymbol{\chi}_{p_\varepsilon}$ should decrease, if multipath ray estimates are properly obtained. The size of $\boldsymbol{\chi}_{p_\varepsilon}$ is equal to the sizes of both $\mathbf{R}_p$ and $\hat{\mathbf{R}}_{p_{\varepsilon-1}}$, and the indices y and z are used to denote the row number and the column number, respectively, for each element in each of these three matrices.

To obtain y_{peak} and z_{peak}, the following operation is executed:

$$(y_{peak}, z_{peak}) = \arg\max_{y,z} \left\{ \left| \boldsymbol{\chi}_{p_\varepsilon}(y,z) \right|_{y=0}^{N_\tau - 1} \right\}_{z=0}^{N_f - 1} \tag{3.49}$$

When multiple decomposition iterations are executed (when $\varepsilon > 0$), there may be a desire to bound initial Doppler frequency and propagation delay parameter outputs to a range with some proximity to the parameter outputs from the first iteration. The intent of this would be to focus the decomposition process on a region within the search grid where decomposition is to be emphasized. When $\varepsilon > 0$, the decomposition outputs $\hat{\mathbf{f}}_{p_0} = \hat{f}_{p_{0_0}}$ and $\hat{\boldsymbol{\tau}}_{p_0} = \hat{\tau}_{p_{0_0}}$ are available to the decomposition processor. These parameter outputs are used in constructing the limits within which initial Doppler frequency and propagation delay estimates are bounded. Parameter estimation is then performed using the following modification to the operation in (3.49):

$$(y_{peak}, z_{peak}) = \arg\max_{y,z} \left[\left\{ \left| \boldsymbol{\chi}_{p_\varepsilon}(y,z) \right|_{y=y_{\min}}^{y_{\max}} \right\}_{z=z_{\min}}^{z_{\max}} \right] \tag{3.50}$$

where

$$y_{\min} = \arg\min_{y} \left[\left| \beta_y - \tau_{\min} \right|_{y=0}^{N_\tau - 1}, \beta_y \geq \tau_{\min} \right] \tag{3.51}$$

$$y_{\max} = \arg\min_{y} \left[\left| \beta_y - \tau_{\max} \right|_{y=0}^{N_\tau - 1}, \beta_y \leq \tau_{\max} \right] \tag{3.52}$$

$$z_{\min} = \arg\min_{z} \left[\left| \gamma_z - f_{\min} \right|_{z=0}^{N_f - 1}, \gamma_z \geq f_{\min} \right] \tag{3.53}$$

$$z_{\max} = \arg\min_{z} \left[\left| \gamma_z - f_{\max} \right|_{z=0}^{N_f - 1}, \gamma_z \leq f_{\max} \right] \tag{3.54}$$

The carrier frequency trial parameter range boundaries are $f_{\min} = \hat{\mathbf{f}}_{p_0} - 1.1/T_{PIT}$ and $f_{\max} = \hat{\mathbf{f}}_{p_0} + 1.1/T_{PIT}$, where T_{PIT} is the pre-detection integration period (PIT). The propagation delay trial parameter range boundaries are $\tau_{\min} = \hat{\boldsymbol{\tau}}_{p_0} - 1.1T_c$ and $\tau_{\max} = \hat{\boldsymbol{\tau}}_{p_0} + 1.1T_c$. These boundaries have been selected to focus the decomposition algorithm on regions within the search grid where, if present, the received GNSS civil signal and multipath content is reasonably assured to be contained. Furthermore, this is the region within the search grid where the impact of multipath has the most consequence, in terms of pseudorange measurements [9].

3.3.7 Estimation of Initial Parameters. This process accepts $\boldsymbol{\chi}_{p_\varepsilon}$, y_{peak}, and z_{peak} as inputs and outputs the initial estimate of the ray waveform amplitude $\tilde{A}_{p_\varepsilon}$, the initial estimate of the Doppler frequency offset $\tilde{f}_{p_\varepsilon}$, the initial estimate of the propagation delay $\tilde{\tau}_{p_\varepsilon}$, and the initial estimate of the carrier phase $\tilde{\phi}_{p_\varepsilon}$. All of these estimates are for the εth ray in ensemble ε.

To obtain initial estimates for the ray waveform amplitude $\tilde{A}_{p_\varepsilon}$, the Doppler frequency offset $\tilde{f}_{p_\varepsilon}$, the propagation delay $\tilde{\tau}_{p_\varepsilon}$, and the carrier phase $\tilde{\phi}_{p_\varepsilon}$, the following four equations are used:

$$\tilde{A}_{p_\varepsilon} = \left|\boldsymbol{\chi}_{p_\varepsilon}(y_{peak}, z_{peak})\right| \tag{3.55}$$

$$\tilde{\tau}_{p_\varepsilon} = \beta_{y_{peak}} \tag{3.56}$$

$$\tilde{f}_{p_\varepsilon} = \gamma_{z_{peak}} \tag{3.57}$$

$$\tilde{\phi}_{p_\varepsilon} = \angle\boldsymbol{\chi}_{p_\varepsilon}(y_{peak}, z_{peak}) \tag{3.58}$$

where $\angle$ denotes the phase angle computation, which is the same as the four quadrant arctangent operation $\tan^{-1}(\text{imag}(\boldsymbol{\chi}_{p_\varepsilon}(y_{peak}, z_{peak}))/\text{real}(\boldsymbol{\chi}_{p_\varepsilon}(y_{peak}, z_{peak})))$.

3.3.8 Decomposition of the Search Space. This process accepts either $\tilde{A}_{p_\varepsilon}$, $\tilde{f}_{p_\varepsilon}$, $\tilde{\tau}_{p_\varepsilon}$, and $\tilde{\phi}_{p_\varepsilon}$ (in cases when $\varepsilon = 0$) or $\tilde{\mathbf{A}}_{p_\varepsilon}$, $\tilde{\mathbf{f}}_{p_\varepsilon}$, $\tilde{\boldsymbol{\tau}}_{p_\varepsilon}$, and $\tilde{\boldsymbol{\phi}}_{p_\varepsilon}$ (in cases when $\varepsilon > 0$). Output from this process are the final ray ensemble parameter estimates associated with $\hat{\mathbf{r}}_{p_\varepsilon}$: the peak amplitude estimates $\hat{\mathbf{A}}_{p_\varepsilon}$, the frequency estimate $\hat{\mathbf{f}}_{p_\varepsilon}$, the propagation delay estimates $\hat{\boldsymbol{\tau}}_{p_\varepsilon}$, and the carrier phase estimates $\hat{\boldsymbol{\phi}}_{p_\varepsilon}$. Estimates are obtained for each of the $\varepsilon + 1$ rays in ensemble ε.

$\tilde{\mathbf{A}}_{p_\varepsilon}$, $\tilde{\mathbf{f}}_{p_\varepsilon}$, $\tilde{\boldsymbol{\tau}}_{p_\varepsilon}$, $\tilde{\boldsymbol{\phi}}_{p_\varepsilon}$, $\hat{\mathbf{A}}_{p_\varepsilon}$, $\hat{\mathbf{f}}_{p_\varepsilon}$, $\hat{\boldsymbol{\tau}}_{p_\varepsilon}$, and $\hat{\boldsymbol{\phi}}_{p_\varepsilon}$ are defined as follows:

$$\tilde{\mathbf{A}}_{p_\varepsilon} = \left[\hat{\mathbf{A}}_{p_{\varepsilon-1}}, \tilde{A}_{p_\varepsilon}\right] \tag{3.59}$$

$$\tilde{\mathbf{f}}_{p_\varepsilon} = \left[\hat{\mathbf{f}}_{p_{\varepsilon-1}}, \tilde{f}_{p_\varepsilon}\right] \tag{3.60}$$

$$\tilde{\boldsymbol{\tau}}_{p_\varepsilon} = \left[\hat{\boldsymbol{\tau}}_{p_{\varepsilon-1}}, \tilde{\tau}_{p_\varepsilon}\right] \tag{3.61}$$

$$\tilde{\boldsymbol{\phi}}_{p_\varepsilon} = \left[\hat{\boldsymbol{\phi}}_{p_{\varepsilon-1}}, \tilde{\phi}_{p_\varepsilon}\right] \tag{3.62}$$

where

$$\hat{\mathbf{A}}_{p_\varepsilon} = \left[\hat{A}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} \tag{3.63}$$

$$\hat{\mathbf{f}}_{p_\varepsilon} = \left[\hat{f}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} \tag{3.64}$$

$$\hat{\boldsymbol{\tau}}_{p_\varepsilon} = \left[\hat{\tau}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} \tag{3.65}$$

$$\hat{\boldsymbol{\phi}}_{p_\varepsilon} = \left[\hat{\phi}_{p_{\varepsilon_m}}\right]_{m=0}^{\varepsilon} \tag{3.66}$$

Figure 3.3 illustrates the flowchart outlining the process used to decompose the search space. This flowchart involves the execution of several subprocesses, corresponding with the sections in this section, where discussion of the specifics of each of these subprocesses is provided. These subprocesses are as follows: defining of the boundaries for the ranges of trial parameter values (Section 3.3.8.1), computation of the search grid for received data that will be used in the decomposition subprocess (Section 3.3.8.2), computation of the trial estimate search grid (Section 3.3.8.3), comparison of the received and trial estimate search grids (Section 3.3.8.4), computation of the error between the received and trial estimate search grids (Section 3.3.8.5), and generation of new trial parameters (Section 3.3.8.6).

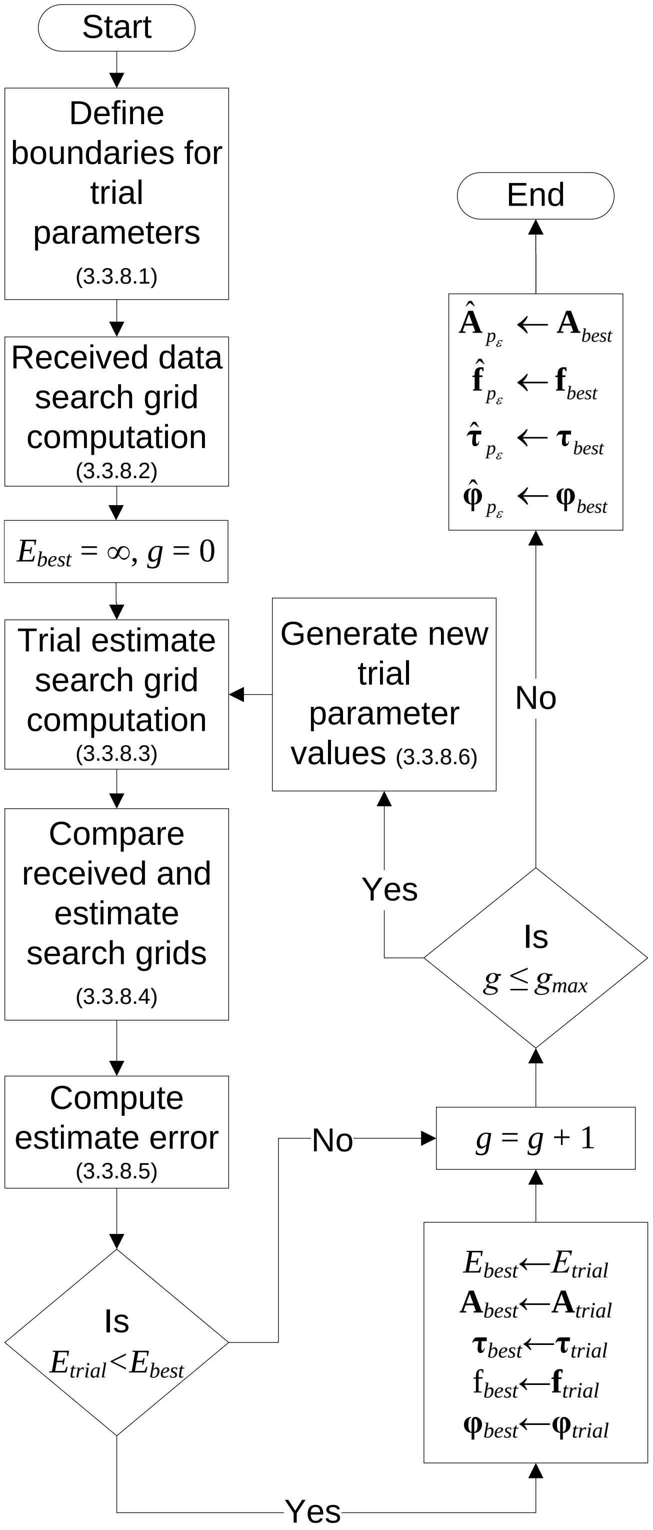
Start
Define boundaries for trial parameters
(3.3.8.1)
Received data search grid computation
(3.3.8.2)
$E_{best} = \infty$, $g = 0$
Trial estimate search grid computation
(3.3.8.3)
Compare received and estimate search grids
(3.3.8.4)
Compute estimate error
(3.3.8.5)
Is
$E_{trial} < E_{best}$
Yes
No
$E_{best} \leftarrow E_{trial}$
$\mathbf{A}_{best} \leftarrow \mathbf{A}_{trial}$
$\boldsymbol{\tau}_{best} \leftarrow \boldsymbol{\tau}_{trial}$
$\mathrm{f}_{best} \leftarrow \mathbf{f}_{trial}$
$\boldsymbol{\varphi}_{best} \leftarrow \boldsymbol{\varphi}_{trial}$
$g = g + 1$
Is
$g \leq g_{max}$
Yes
No
Generate new trial parameter values (3.3.8.6)
$\hat{\mathbf{A}}_{p_\varepsilon} \leftarrow \mathbf{A}_{best}$
$\hat{\mathbf{f}}_{p_\varepsilon} \leftarrow \mathbf{f}_{best}$
$\hat{\boldsymbol{\tau}}_{p_\varepsilon} \leftarrow \boldsymbol{\tau}_{best}$
$\hat{\boldsymbol{\varphi}}_{p_\varepsilon} \leftarrow \boldsymbol{\varphi}_{best}$
End

Figure 3.3: Flowchart illustrating the search space decomposition process. Processes are outlined in this section, with exceptions in cases where the flowchart process label is self-explanatory, such as "$g = g + 1$." The section number for the discussion corresponding with each block is provided in the flowchart.

3.3.8.1 Define Boundaries for Trial Parameters. This subprocess accepts either $\tilde{A}_{p_0}$, $\tilde{f}_{p_0}$, $\tilde{\tau}_{p_0}$, and $\tilde{\phi}_{p_0}$ (in cases when $\varepsilon = 0$) or $\hat{\mathbf{A}}_{p_0}$, $\hat{\mathbf{f}}_{p_0}$, $\hat{\boldsymbol{\tau}}_{p_0}$, and $\hat{\boldsymbol{\phi}}_{p_0}$ (in cases when $\varepsilon > 0$). Output from this process are boundary values (both minimum and maximum) for the trial parameters used to generate the trial ray waveforms being considered. Parameter boundaries are output for the peak amplitude ($A_{\min}$ and $A_{\max}$), the carrier frequency ($f_{\min}$ and $f_{\max}$), the propagation delay ($\tau_{\min}$ and $\tau_{\max}$), and the carrier phase ($\phi_{\min}$ and $\phi_{\max}$). Also made available is N_p.

This subprocess makes use of the same boundaries as those used for the SAGE-based decomposition process, and are defined in Section 2.6.3.1. The boundaries for $\mathbf{f}_{trial}$ and $\boldsymbol{\tau}_{trial}$ defined in (2.47) - (2.48) are applied in this subprocess as well, and are restated in this section of the dissertation for convenience.

This subprocess is used to simply define the ranges from which trial parameters for the peak amplitude $\mathbf{A}_{trial}$, the carrier frequency $\mathbf{f}_{trial}$, the propagation delay $\boldsymbol{\tau}_{trial}$, and the carrier phase $\boldsymbol{\phi}_{trial}$ in the decomposition process may be obtained. These ranges are defined as follows:

$$\mathbf{A}_{trial} \in [A_{\min}, A_{\max}] \tag{3.67}$$

$$\mathbf{f}_{trial} \in [f_{\min}, f_{\max}] \tag{3.68}$$

$$\boldsymbol{\tau}_{trial} \in [\tau_{\min}, \tau_{\max}] \tag{3.69}$$

$$\boldsymbol{\phi}_{trial} \in [\phi_{\min}, \phi_{\max}] \tag{3.70}$$

where trial parameter vectors $\mathbf{A}_{trial}$, $\mathbf{f}_{trial}$, $\boldsymbol{\tau}_{trial}$, and $\boldsymbol{\phi}_{trial}$ generated using simulated annealing are expressed as follows:

$$\mathbf{A}_{trial} = [A_{trial_m}]_{m=0}^{\varepsilon} \tag{3.71}$$

$$\mathbf{f}_{trial} = [f_{trial_m}]_{m=0}^{\varepsilon} \tag{3.72}$$

$$\boldsymbol{\tau}_{trial} = [\tau_{trial_m}]_{m=0}^{\varepsilon} \tag{3.73}$$

$$\boldsymbol{\phi}_{trial} = [\phi_{trial_m}]_{m=0}^{\varepsilon} \tag{3.74}$$

In this research, these boundaries are defined as follows: $A_{\min} = 0$ and $A_{\max} = \max\{|\mathbf{r}_p|\}$. This range is set to simply dictate in generation of trial parameters that the peak amplitude of the ray waveform must not be outside the range of possible values found in the received data vector. Because of the signal to noise ratio of GPS data recorded using a software receiver, this range is appropriate, since peak noise amplitude values will almost certainly exceed the peak amplitude of a GPS waveform. For some simulation circumstances, however, $A_{\max}$ may need to be modified. The carrier frequency parameter range boundaries and the propagation delay parameter range boundaries are defined previously in Section 3.3.7. These same boundaries are employed for this subprocess. The carrier frequency trial parameter range boundaries are $f_{\min} = \tilde{f}_{p0} - 1.1/T_{PIT}$ and $f_{\max} = \tilde{f}_{p0} + 1.1/T_{PIT}$ (in cases when $\varepsilon = 0$) or $f_{\min} = \hat{\mathbf{f}}_{p0} - 1.1/T_{PIT}$ and $f_{\max} = \hat{\mathbf{f}}_{p0} + 1.1/T_{PIT}$ (in cases when $\varepsilon > 0$). In cases when $\varepsilon > 0$, $f_{\min}$ and $f_{\max}$ are scalars, since $\hat{\mathbf{f}}_{p0} = \hat{f}_{p0_0}$ (only one ray waveform is parameterized in the estimate waveform generated from the initial ensemble, since the highest ray index for an ensemble is equal to the ensemble index itself). The propagation delay trial parameter range boundaries are $\tau_{\min} = \tilde{\tau}_{p0} - 1.1T_c$ and $\tau_{\max} = \tilde{\tau}_{p0} + 1.1T_c$ (in cases when $\varepsilon = 0$) or $\tau_{\min} = \hat{\boldsymbol{\tau}}_{p0} - 1.1T_c$ and $\tau_{\max} = \hat{\boldsymbol{\tau}}_{p0} + 1.1T_c$ (in cases when $\varepsilon > 0$). In cases when $\varepsilon > 0$, $\tau_{\min}$ and $\tau_{\max}$ are scalars, since $\hat{\boldsymbol{\tau}}_{p0} = \hat{\tau}_{p0_0}$. The carrier phase trial parameter range boundaries are $\phi_{\min} = -2\pi$ and $\phi_{\max} = 2\pi$. This range is chosen to ensure the true value of the carrier phase falls in the range, and to account for the circular, rather than linear, nature of carrier phase values (a carrier phase of $-\pi$ is equal to a carrier phase of π).

3.3.8.2 Received Data Search Grid Computation. This subprocess accepts $\mathbf{r}_p$ and N_p as inputs and outputs the search grid to be used in the search space decomposition $\mathbf{R}_s$.

Note that $\mathbf{R}_s$ is distinct from $\mathbf{R}_p$. This is a design choice that has been made for this research, as the initial estimate used in the search space decomposition process is required to be relatively accurate, thus the use of tight grid spacings. The search grid $\mathbf{R}_s$ is used to obtain the estimate for the key parameters in an iterative process. The tighter the grid spacing of the search grid, the greater the computational requirement. Therefore, the construction of these two search grids is decoupled to ensure the simultaneous accuracy of initial estimates while still retaining the ability to perform the search space decomposition in a reasonable timeframe.

The search grid $\mathbf{R}_s$ used in the stochastic search process is computed as follows:

$$\mathbf{R}_s = \text{grid}\Big(\mathbf{r}_p, \tau_{window}, f_{window}, \Delta\tau_{input}, [\Delta f_{input}]_{N_{fft}=N_{f_{input}}}\Big) \tag{3.75}$$

where $\tau_{window} = 2T_c$ (as before), $f_{window} = 10$ kHz (as before), $\Delta\tau_{input} = T_s$ (as before), and the frequency grid spacing Δf_s is such that the number of points in the FFT used to compute the search grid is equal to $N_{f_{input}}$. The equation for $N_{f_{input}}$ is as follows:

$$N_{f_{input}} = 2^{\lambda}, \lambda = \left\lceil \log_2\left(\max\left(\frac{f_s}{N_p}, \frac{2}{3N_pT_s}\right)\right)\right\rceil \tag{3.76}$$

Note that (3.36) is used in the computation of Δf_s. The number of code delay values, thus the number of rows in the search grid, is denoted N_{τ_s}. The number of frequency values, thus the number of columns in the search grid, is denoted N_{f_s}.

An additional self-explanatory subprocess following the computation of the received data search grid is the initialization of the trial iteration index $g = 0$ and the best error computation $E_{best} = \infty$.

3.3.8.3 Compute Trial Estimate Search Grid. This process accepts $\mathbf{A}_{trial}$, $\mathbf{f}_{trial}$, $\boldsymbol{\tau}_{trial}$, and $\boldsymbol{\phi}_{trial}$ as inputs and outputs the search grid for the trial estimate $\mathbf{R}_{trial}$.

In the original publication outlining SDPA [141], the search grid was generated through the computation of the discrete time waveform $\mathbf{r}_{trial}$ and the direct transformation of $\mathbf{r}_{trial}$ to the search grid $\mathbf{R}_{trial}$. However, this method is relatively computationally intensive, and is dramatically computationally outpaced by modifying a search space-domain model found in literature [145] that can be used to obtain an approximation to $\mathbf{R}_{trial}$ that follows:

$$\begin{aligned}\mathbf{R}_{trial} &\approx \text{grid}\left(\mathbf{r}_{trial}, \tau_{window}, f_{window}, \Delta\tau_{input}, \Delta f_{input}\right)\\ &\approx \begin{bmatrix} R_{trial}\left(\beta_0,\gamma_0\right) & R_{trial}\left(\beta_0,\gamma_1\right) & \cdots & R_{trial}\left(\beta_0,\gamma_{N_f-1}\right)\\ R_{trial}\left(\beta_1,\gamma_0\right) & R_{trial}\left(\beta_1,\gamma_1\right) & \cdots & R_{trial}\left(\beta_1,\gamma_{N_f-1}\right)\\ \vdots & \vdots & \ddots & \vdots\\ R_{trial}\left(\beta_{N_\tau-1},\gamma_0\right) & R_{trial}\left(\beta_{N_\tau-1},\gamma_1\right) & \cdots & R_{trial}\left(\beta_{N_\tau-1},\gamma_{N_f-1}\right)\end{bmatrix}\end{aligned} \tag{3.77}$$

where

$$R_{trial}\left(\beta_y,\gamma_z\right) = \sum_{m=0}^{\varepsilon}\left[A_{trial_m}\text{sinc}\left(m,z\right)\varrho(m,y)\exp\left(j(\pi\left(f_{trial_m}-\gamma_z\right)T_{PIT}+\phi_{trial_m}))\right] \tag{3.78}$$

$$\text{sinc}\left(m,z\right) = \left\{\begin{array}{c}\frac{\sin\left(\pi(f_{trial_m}-\gamma_z)T_{PIT}\right)}{\pi(f_{trial_m}-\gamma_z)T_{PIT}}, f_{trial_m}-\gamma_z \neq 0\\ 1, f_{trial_m}-\gamma_z = 0\end{array}\right\} \tag{3.79}$$

$$\varrho(m,y) \approx \left\{\begin{array}{c}1-\left|\frac{\tau_{trial_m}-\beta_y}{T_c}\right|, |\tau_{trial_m}-\beta_y| \leq T_c\\ 0, |\tau_{trial_m}-\beta_y| > T_c\end{array}\right\} \tag{3.80}$$

The same grid parameters used to define the received search grid $\mathbf{R}_s$ in (3.75) are used to define the trial estimate search grid $\mathbf{R}_{trial}$ in (3.77).

3.3.8.4 Compare Received and Estimate Search Grids. This subprocess accepts $\mathbf{R}_s$ and $\mathbf{R}_{trial}$ as inputs and outputs the comparison matrix $\boldsymbol{\chi}_{trial}$. $\boldsymbol{\chi}_{trial}$ is computed using the following equation:

$$\boldsymbol{\chi}_{trial} = \mathbf{R}_s - \mathbf{R}_{trial} \tag{3.81}$$

3.3.8.5 Compute Estimate Error. This subprocess accepts $\boldsymbol{\chi}_{trial}$ as an input and outputs the trial estimate error E_{trial} associated with the trial parameters.

E_{trial} is computed using the following equation:

$$E_{trial} = \sqrt{\frac{1}{N_{\tau_s} N_{f_s}} \sum_{y=0}^{N_{\tau_s}-1} \sum_{z=0}^{N_{f_s}-1} \left| \boldsymbol{\chi}_{trial}(y,z) \right|^2} \tag{3.82}$$

The decision point "Is $E_{trial} < E_{best}$" is used to determine whether or not the current error E_{trial} is superior to previous error values. If this is the case, the parameters for the best estimate are asserted as the current parameters in the subsequent subprocess, denoted "$E_{best} \leftarrow E_{trial}$, $\mathbf{A}_{best} \leftarrow \mathbf{A}_{trial}$, $\boldsymbol{\tau}_{best} \leftarrow \boldsymbol{\tau}_{trial}$, $\mathbf{f}_{best} \leftarrow \mathbf{f}_{trial}$, $\boldsymbol{\phi}_{best} \leftarrow \boldsymbol{\phi}_{trial}$".

Regardless of whether or not $E_{trial} < E_{best}$, the iteration index g is then incremented in the process that follows these operations, and is denoted "$g = g + 1$".

The decision point "Is $g \leq g_{\max}$" is used to determine whether further iterations of trial parameter generation and comparison with received data shall take place. If $g \leq g_{\max}$, then further iterations occur, starting with the generation of new trial parameter values, as outlined in Section 3.3.8.6. Otherwise, the best estimates are asserted as the final parameters. In this process, the number of iterations used is typically $g_{\max} = 1000$ when using simulated annealing to generate new trial parameter values, though $g_{\max} = 1000$ is an arbitrary value typically used for the number of iterations in generic simulated annealing literature. The use of a ceiling on the number of trial parameter iterations permitted the algorithm is partially a particular quality of simulated annealing, but capping the maximum number of permitted iterations is usually available to other stochastic optimization methods.

The process to assert the best estimates as the final estimates is denoted "$\hat{\mathbf{A}}_{p_\varepsilon} \leftarrow \mathbf{A}_{best}$, $\hat{\mathbf{f}}_{p_\varepsilon} \leftarrow \mathbf{f}_{best}$, $\hat{\boldsymbol{\tau}}_{p_\varepsilon} \leftarrow \boldsymbol{\tau}_{best}$, $\hat{\boldsymbol{\phi}}_{p_\varepsilon} \leftarrow \boldsymbol{\phi}_{best}$". Note that final estimate values will always be output from the search space decomposition process, regardless of the quality of the estimate. Upon assertion of final parameter values, the search space decomposition process is complete.

3.3.8.6 Generate New Trial Parameter Values. This subprocess accepts g as an input and outputs new values of $\mathbf{A}_{trial}$, $\mathbf{f}_{trial}$, $\boldsymbol{\tau}_{trial}$, and $\boldsymbol{\phi}_{trial}$. There are different means by which to compute trial parameters, depending on the stochastic search and optimization method used. This research makes use of simulated annealing for selection of trial parameters. Section 2.5.1 discusses the theory behind simulated annealing. Simulated annealing is made available to users practically in the Matlab global optimization toolbox, and has been employed in this research to serve as the computational backbone for trial parameter generation. Trial parameter values generated using the simulated annealing function are output for computation of the trial estimate search grid, discussed in Section 3.3.8.3.

3.3.9 Computation of the Estimate Search Space. This process accepts $\hat{\mathbf{A}}_{p_\varepsilon}$, $\hat{\mathbf{f}}_{p_\varepsilon}$, $\hat{\boldsymbol{\tau}}_{p_\varepsilon}$, and $\hat{\boldsymbol{\phi}}_{p_\varepsilon}$ as inputs and outputs the estimate search grid $\hat{\mathbf{R}}_{p_\varepsilon}$. To compute $\hat{\mathbf{R}}_{p_\varepsilon}$, the estimate discrete time-domain waveform $\hat{\mathbf{r}}_{p_\varepsilon}$ must first be computed. To do this, the following equation is used:

$$\hat{\mathbf{r}}_{p_\varepsilon} = \sum_{m=0}^{\varepsilon} \hat{\mathbf{r}}_{p_{\varepsilon_m}} \tag{3.83}$$

where

$$\hat{\mathbf{r}}_{p_{\varepsilon_m}} = \begin{cases} \hat{A}_{p_{\varepsilon_m}} \hat{\mathbf{x}}_{p_{\varepsilon_m}} \circ \left[\exp\left(j\left(2\pi(f_{\mathrm{IF}} + \hat{f}_{p_{\varepsilon_m}})T_s w + \hat{\phi}_{p_{\varepsilon_m}}\right)\right)\right]_{w=0}^{N_p-1}, & \mathbf{r}_p \in \mathbb{C} \\ \mathrm{Re}\left\{\hat{A}_{p_{\varepsilon_m}} \hat{\mathbf{x}}_{p_{\varepsilon_m}} \circ \left[\exp\left(j\left(2\pi(f_{\mathrm{IF}} + \hat{f}_{p_{\varepsilon_m}})T_s w + \hat{\phi}_{p_{\varepsilon_m}}\right)\right)\right]_{w=0}^{N_p-1}\right\}, & \mathbf{r}_p \in \mathbb{R} \end{cases} \tag{3.84}$$

and the computation of $\hat{\mathbf{x}}_{p_{\varepsilon_m}}$ makes use of (3.40) as follows:

$$\hat{\mathbf{x}}_{p_{\varepsilon_m}} = \mathbf{x}_s\left(\hat{\tau}_{p_{\varepsilon_m}}\right) \tag{3.85}$$

To compute the search grid $\hat{\mathbf{R}}_{p_\varepsilon}$, the following equation is used:

$$\hat{\mathbf{R}}_{p_\varepsilon} = \mathrm{grid}(\hat{\mathbf{r}}_{p_\varepsilon}, \tau_{window}, f_{window}, \Delta\tau_{input}, \Delta f_{input}) \tag{3.86}$$

3.3.10 Comparison of Received and Estimate Search Spaces. This process accepts $\mathbf{R}_p$ and $\hat{\mathbf{R}}_{p_\varepsilon}$ as inputs and outputs the comparison matrix $\boldsymbol{\chi}_{p_\varepsilon}$.

The equation used to compute $\boldsymbol{\chi}_{p_\varepsilon}$ is as follows:

$$\boldsymbol{\chi}_{p_\varepsilon} = \mathbf{R}_{p_\varepsilon} - \hat{\mathbf{R}}_{p_\varepsilon} \tag{3.87}$$

This process is executed distinctly from the comparison subprocess described in Section 3.3.8.4 as part of the search space decomposition process. While the result of the comparison process in Section 3.3.8.4 would yield a similar result, the process described above in this section is different in that the search grids denoted $\boldsymbol{\chi}_{p_\varepsilon}$, $\mathbf{R}_{p_\varepsilon}$, and $\hat{\mathbf{R}}_{p_\varepsilon}$ may use a different propagation delay grid spacing $\Delta\tau$ and a different frequency grid spacing Δf from that used in Section 3.3.8.4.

3.3.11 Computation of the Estimate Error. This process accepts $\boldsymbol{\chi}_{p_\varepsilon}$ as an input and outputs the estimate error E_{p_ε}. The error E_{p_ε} is computed as follows:

$$E_{p_\varepsilon} = \sqrt{\frac{1}{N_\tau N_f} \sum_{y=0}^{N_\tau - 1} \sum_{z=0}^{N_f - 1} \left| \boldsymbol{\chi}_{p_\varepsilon}(y, z) \right|^2} \tag{3.88}$$

The decision point "Is $\varepsilon = 0$" is used to determine whether the first, or LOS, ray is being considered. If so, the estimate parameters output from the search space decomposition are taken as is to be the estimate parameters for the LOS ray. If $\varepsilon > 0$, the algorithm advances to another decision point, denoted "Is $E_{p_\varepsilon} \leq E_{p_{\varepsilon-1}}$". This decision is made to determine if the current ensemble of ray parameter estimates improves the global estimate that makes use of all ray parameters within ε obtained using the decomposition algorithm. Therefore, if the parameter estimates associated with ε improve or remain the same in error relative to the estimates associated with ensemble $\varepsilon - 1$, the ensemble is declared to be a valid parameterization, and the ray parameters are output to the ray database, as discussed in Section 3.3.12. If the ensemble does not reduce or match the error from the previous ensemble, then the ensemble is not considered to be valid, the ensemble is discarded, and other regional maxima are considered, as discussed in Section 3.3.14.

3.3.12 Output of Parameters to the Ray Database. This process accepts $\hat{\mathbf{A}}_{p_\varepsilon}$, $\hat{\mathbf{f}}_{p_\varepsilon}$, $\hat{\boldsymbol{\tau}}_{p_\varepsilon}$, and $\hat{\boldsymbol{\phi}}_{p_\varepsilon}$ as inputs and does not output anything. The ray database is used to simply store estimate parameters for use in inter-PIT algorithms. The whole point of the decomposition algorithm is to extract these parameters, eventually yielding this database. Using this database, multipath rays present in more than one PIT may be studied.

3.3.13 Computation of the Stopping Criteria Statistic. This process accepts $\boldsymbol{\chi}_{p_\varepsilon}$ as an input and outputs the stopping criteria statistic F_{p_ε}. F_{p_ε} can be described as a peak signal content magnitude to noise power comparison parameter. The methodology behind employment of the stopping criteria is available in [142], but is explained in this section as well. F_{p_ε} is computed using the cumulative distribution function (cdf) of the Rayleigh distribution as follows [146]:

$$F_{p_\varepsilon} = 1 - \exp\left(-\frac{\xi_{p_\varepsilon}^2}{2\sigma_\chi^2}\right), \xi \geq 0 \tag{3.89}$$

To obtain ξ_{p_ε}, (3.50) is used to determine the initial Doppler frequency and propagation delay parameters, where $y_{\min}$ and $y_{\max}$ bind the indices considered in $\boldsymbol{\chi}_{p_\varepsilon}$ to those corresponding with the following condition: $\beta_{y_{\min}} \leq \hat{\tau}_{p0_0} \leq \beta_{y_{\max}}$. $z_{\min}$ and $z_{\max}$ bind the indices considered in establishing ξ_{p_ε} to those corresponding with the following condition: $\gamma_{z_{\min}} \leq \hat{f}_{p0_0} \leq \gamma_{z_{\max}}$. ξ_{p_ε} is then obtained using the following equation:

$$\xi_{p_\varepsilon} = \left|\boldsymbol{\chi}_{p_\varepsilon}\left(y_{peak}, z_{peak}\right)\right| \tag{3.90}$$

The carrier frequency trial parameter range boundaries are $f_{\min} = \hat{\mathbf{f}}_{p0} - 1.1/T_{PIT}$ and $f_{\max} = \hat{\mathbf{f}}_{p0} + 1.1/T_{PIT}$, same as those used in Section 3.3.7. The propagation delay trial parameter range boundaries are $\tau_{\min} = \hat{\boldsymbol{\tau}}_{p0} - (T_c + 9 \times 10^{-7})$ and $\tau_{\max} = \hat{\boldsymbol{\tau}}_{p0} + (T_c +$

9×10^{-7}). Multipath signal content is expected to be delayed at most by 1.8×10^{-6} sec of the LOS ray, per findings outlined in [70] and applied per Section 4.2.2.5, thus the use of this value in setting the boundary conditions for the propagation delay.

To obtain σ_{χ}, it is assumed that there is no signal content present outside the boundary conditions outlined in the previous paragraph. The complex values of all the elements in $\boldsymbol{\chi}_{p_\varepsilon}$ from grid points with indices corresponding with Doppler frequency or propagation delay values outside these boundaries are concatenated into a single vector. The standard deviation σ_{χ} of either the real or imaginary components is then computed. Given the assumption that this noise content originates from circular complex AWGN, the standard deviation of the real components should be equal to the standard deviation of the imaginary components.

Upon computation of F_{p_ε}, the stopping criteria is then applied using the following decision criteria:

$$F_{p_\varepsilon} \quad \overset{\text{Continue}}{\underset{\text{Halt}}{\gtrless}} \quad \Upsilon \tag{3.91}$$

where Υ is the stopping criteria threshold. The stopping criteria threshold Υ is determined through simulation. The results of simulation leading to the determination of Υ are outlined in Chapter IV. If $F_{p_\varepsilon} \geq \Upsilon$, the ensemble index ε is incremented (using the "$\varepsilon = \varepsilon + 1$" process), and then the next ensemble for the same PIT is considered for decomposition, starting with estimation of the initial search grid peak location discussed in Section 3.3.6. If $F_{p_\varepsilon} < \Upsilon$, decomposition of the current PIT is halted, the PIT index is incremented ($p = p + 1$), and the algorithm advances to the next PIT. If there are no more integration periods remaining to be processed ($p > p_{\max}$, where $p_{\max}$ is the PIT index of the last PIT to be processed in the data), the algorithm is then halted.

3.3.14 Determination of Alternate Regional Maxima. This process accepts $\boldsymbol{\chi}_{p_{\varepsilon-1}}$ as an input and outputs y_{alt} and z_{alt}. The absolute values of each of the elements in $\boldsymbol{\chi}_{p_{\varepsilon-1}}$ are evaluated to determine the locations of regional maxima. The Matlab `imregionalmax` command can be used to determine these locations in software, and the 8-neighborhood regional maxima are determined. If there are regional maxima within $\boldsymbol{\chi}_{p_{\varepsilon-1}}$ that have not yet been considered for estimation, then the values of y_{alt} and z_{alt} corresponding with the search grid locations where these peaks are found are output from this process. The algorithm then continues with estimation of alternate search grid peak locations, discussed in Section 3.3.15. Otherwise, the PIT index is incremented ("$p = p+1$") and the next PIT is decomposed. If "$p > p_{\max}$", there are no more PIT-sized vectors of received data to consider, and the algorithm is completed.

3.3.15 Estimation of Alternate Search Grid Peak Locations. This process accepts $\boldsymbol{\chi}_{p_{\varepsilon-1}}$, y_{alt}, and z_{alt} as inputs and outputs y_{peak} and z_{peak}. These output row and column indices for the search grid correspond with the element in $\boldsymbol{\chi}_{p_{\varepsilon-1}}$ that has the highest magnitude. Upon output of y_{peak} and z_{peak}, initial parameter estimates making use y_{peak} and z_{peak} take place, as found in Section 3.3.7. There may again be a desire to bind values of y_{peak} and z_{peak} inside a boundary, as discussed in Section 3.3.6. In this case, (3.50) is once again used to limit y_{peak} and z_{peak} to values defined by boundaries discussed in Section 3.3.6.

3.3.16 Algorithm Summary. This section outlines the algorithm used in this research to decompose a GPS L1 C/A-coded signal with interference and noise added to it that may be recorded by an RF receiver front end and made available for use in a software receiver.

3.4 Comparison of Multipath Estimation Algorithms

Section 3.4.1 outlines similarities and differences between SDPA, SAGE, and MEDLL, with emphasis on the details of the contrasts between SDPA and the SAGE-based algorithm. Section 3.4.2 discusses the motivation for developing SDPA despite the prior development of the SAGE-based algorithm.

Table 3.1: Comparison of MEDLL, SAGE, and SDPA characteristics

Comparison point	MEDLL	SAGE	SDPA
Comparison domain	Time [114]	Time [57]	Time-frequency
Uses ML estimation	Yes [114]	Yes [57]	No
Uses tracking loops	Yes [114]	Optional [102]	No
Estimates ray AOA	Yes [147]	Yes [104]	No
Ray count (M) considered	Yes [114]	No	Yes
Known to be used commercially	Yes [110]	No	No
Estimates all parameters simultaneously	Yes [54]	No [57]	Yes

3.4.1 Algorithm Comparison. Among GNSS receiver signal processing techniques that explicitly estimate multipath ray waveform parameters to measure or mitigate channel characteristics, SDPA is believed to be most closely comparable to MEDLL and SAGE. Therefore, it is appropriate to explicitly consider comparison and contrast points as part of the research outlined in this dissertation. Table 3.1 provides a comparison of MEDLL, SAGE, and SDPA.

All three of these GNSS receiver signal processing techniques explicitly estimate direct path and multipath ray waveform parameters including the waveform amplitude, the Doppler frequency, the relative propagation delay, and the carrier phase. For all three techniques, these estimates are obtained on an integration period-by-integration period basis.

All three of these techniques make use of some sort of interference cancellation feature, where each of the ray waveforms present in received data are explicitly accounted for, and may be deducted from the received waveform to obtain an estimate of the sum of the direct path ray waveform and the received noise.

All three of these techniques make use of some variation on the following iterative sequence: 1) initialization, 2) cancellation of successive multipath correlations, and 3) convergence (repeating step two until a convergence criteria is satisfied) [57] [110].

MEDLL fundamentally relies upon the estimation of ray waveform parameters through the minimization of the correlation output (or RMS error) between a sum of ray waveform estimates (reference signals) and the received signal [54]. Solving for ray waveform parameter estimates is completed by setting the partial derivatives (with respect to the parameter of interest, be it ray waveform amplitude, propagation delay, or initial carrier phase of ray waveform equations) equal to zero [54].

As discussed in Section 2.6.1, SAGE relies upon the iterative use of the E-step and the M-step until parameter estimates converge. individual parameters for individual rays are serially considered as part of the SAGE algorithm, as opposed to the use of partial derivatives found in MEDLL.

The comparison domain is the domain where received and reference waveform comparisons occur. In the case of MEDLL and SAGE, waveform comparison in the form of correlation between received and reference signals occurs in the discrete time domain. In the case of SDPA, comparison between received and reference signals occurs in the time-frequency search space ("search grid").

The use of ML estimation provides parameter estimates that maximize the joint PDF of sample data. Because of this, ML estimation provides the optimal means for estimating parameters in the absence of prior knowledge [110]. Both MEDLL and SAGE make use of ML estimation, whereas SDPA uses stochastic optimization (with simulated annealing) to generate trial parameter values.

MEDLL explicitly uses a DLL tracking loop for every path, either direct path or multipath, to generate propagation delay estimates in parallel. Use of tracking loops has been considered for use in SAGE as well. The use of tracking loops generally improves parameter estimation, but making use of feedback from previous results in the computation of current estimates may bias current and future loop outputs erroneously (in the case where past measurements were poorly estimated).

The use of MEDLL to directly perform estimation of the angle of arrival (AOA, in azimuth and elevation as received using an antenna array) of particular direct path and multipath rays as part of a multipath mitigation methodology is presented in [147]. The use of SAGE in a GNSS receiver where the angles of arrival of individual multipath rays are estimated has been considered, and found to improve multipath parameter estimation performance [104]. This aspect of receiver processing has not been considered for SDPA.

All three of these techniques require that for the integration period of interest, the number of rays be specified. Therefore, there must be some means established by which the number of multipath rays is provided to all of these algorithms. For MEDLL, the maximum likelihood estimate of the number of multipath rays can be computed. For SDPA, a stopping criteria is researched which can be applied to both SDPA and SAGE. More background into this requirement is provided in Section 2.8.3.

For MEDLL and SDPA, all ray waveform parameter estimates are obtained simultaneously in processing. With SAGE, the propagation delay is first obtained, then the Doppler frequency, and then the ray waveform amplitude and initial carrier phase are essentially obtained simultaneously (though the order between the propagation delay and the Doppler frequency can be varied).

As discussed in Section 2.7, MEDLL requires the use of a bank of correlators integrated into delay locked loops by definition. The presence of these loops yields the possibility that estimation errors from prior correlation operations may yield errors in the current estimation process. The presence of these loops works generally to reduce errors in the MEDLL architecture, but the intent of SDPA is to have a parameter estimation algorithm that does not rely upon estimates from prior iterations in the estimation of parameters being considered presently within the architecture. Therefore, further comparison between SDPA and MEDLL is abandoned in favor of limiting comparisons to those between SDPA and SAGE.

The flowcharts illustrated in Figures 2.1 and 3.3 for SAGE and SDPA, respectively, highlight the similarities and difference between the two approaches to signal decomposition. Both SAGE and SDPA make use of some sort of interference cancellation feature. In the case of SAGE, interference cancellation takes place when interfering ray waveforms are subtracted from received data (Section 2.6.3.2). For

SDPA, estimate rays are summed so that every multipath ray for the ensemble is considered simultaneously. These approaches are equivalent, so SAGE and SDPA are similar in this regard.

Ray waveforms are processed sequentially in SAGE, whereas they are processed simultaneously in SDPA. SAGE makes use of discrete time-domain correlation, whereas error values between time-frequency-domain transforms are computed in SDPA. Parameters defining individual ray waveforms within a signal are obtained sequentially in SAGE, whereas in SDPA parameters are obtained simultaneously. Processing is halted when using SAGE when every parameter value for every estimate ray waveform converges. In SDPA, processing is halted when either the stopping criteria is satisfied or the error between the received and estimate search grids cannot be further reduced, regardless of whether individual waveform parameters converge.

The most important point of contrast between SAGE and SDPA, the feature that underpins the premise of the contribution to the body of knowledge in GNSS multipath modeling made with this document, is that SAGE considers sequential instances of correlation, with each instance using a single Doppler frequency value and a single relative propagation delay value. This is equivalent to evaluating only one discrete location in the search grid at a time. SDPA, on the other hand, forces consideration of multiple points between received and estimate search grids with each comparison of search grids. The received search grid will contain the frequency transform of AWGN, and this noise yields uncertainty in amplitude values resulting from correlation in SAGE. The means by which SAGE obtains ray waveform amplitude estimates does nothing to account for this uncertainty, as described in Section 2.6.3.5. This uncertainty is moderated in SDPA. Therefore, in noisy environments, SDPA will outperform SAGE in the reduction of error between true and estimate search grids in specific cases. This insight spurs the research outlined in this document. This distinction holds true between SDPA and MEDLL as well, as MEDLL is similar to SAGE in this regard.

3.4.2 Motivation for Developing an Alternative to the SAGE-based Decomposition Technique. The motivation for developing SDPA is to make use of the time-frequency signal processing capability made possible through the use of search spaces. The search space intrinsically serves as a parallel correlator, whereas algorithms such as SAGE make iterative use of the output of serial correlation operations. Content within the search space is equivalent to correlator outputs for given Doppler frequency and propagation delay offset values. Therefore, a parallel correlation algorithm like SDPA makes use of much more information about the received signal within a given iteration than a serial correlation algorithm such as SDPA.

There will be a trade-off in the use of a parallel versus serial correlation operation. The peak-seeking nature of a serial correlator like SAGE lends to the algorithm avoiding parts of the search space where the signal to noise ratio is relatively low. This avoidance means that signal content that has been corrupted by noise will not impact parameter estimation. A parallel correlation algorithm like SDPA, on the other hand, makes use of areas where the signal to noise ratio for a region within the search space is low. Furthermore, SDPA is making use of more information than SAGE for a given iteration. Making use of this information means that averaging operations are conducted transparently within a parallel correlation operation. This averaging creates a robustness of its own in instances when the signal to noise ratio is poor throughout the search space. Therefore, it's expected that SAGE-based search space estimation will be superior to SDPA in cases when C/N_0 is high over a fixed integration period and that SDPA will outperform SAGE in cases when C/N_0 is low.

This phenomenon can be shown analytically. Assume the received signal model presented in (3.20) is applied in an arbitrary scenario with $\mathbf{r}_p \in \mathbb{R}$ and $M_p = 0$ (so no multipath is present). This model makes use of an additive noise term $\boldsymbol{\eta}_p$ described in (3.22). Assume $\boldsymbol{\eta}_p \in \mathbb{R}$. Furthermore, $\eta[n] \sim N(0, \sigma_{\text{IF}}^2)$. The fundamental equation relating the carrier to noise power ratio C/N_0 to the ray amplitude A_{p_0}, the standard deviation of the noise σ_{IF}, and the sampling frequency f_s is as follows [143]:

$$\frac{C}{N_0} = \frac{A_{p_0}^2 f_s}{4\sigma_{\mathrm{IF}}^2} \tag{3.92}$$

Now solving for A_{p_0}:

$$A_{p_0} = 2\sigma_{\mathrm{IF}}\sqrt{\frac{C/N_0}{f_s}} \tag{3.93}$$

Given these equation, if σ_{IF} is arbitrarily fixed to a value equal to one and f_s is arbitrarily fixed to a value of 3.42 MHz (which is the intermediate frequency of the TRIGR receiver downconverted by four), A_{p_0} can then be calculated for various values of C/N_0. Furthermore, given A_{p_0}, the highest possible magnitude for signal content within the search space is $A_{p_0}/2$ for $\mathbf{r}_p \in \mathbb{R}$ [2]. This value is also equal to the highest possible amplitude computation for SAGE in the absence of noise, as described in Section 2.6.3.5.

The Fourier transform of bandlimited AWGN $\boldsymbol{\eta}_p$ is a complex normally distributed signal [143]. The standard deviation of the real portion of the Fourier transform of $\boldsymbol{\eta}_p$, σ_n, is equal to the standard deviation of the imaginary portion of the Fourier transform of $\boldsymbol{\eta}_p$. The relationship between σ_n and σ_{IF} is as follows [143]:

$$\sigma_n = \sqrt{\frac{\sigma_{\mathrm{IF}}^2}{2N_p}} \tag{3.94}$$

The magnitude of a complex normally distributed signal is Rayleigh distributed [14], so $\mathfrak{F}\left(\boldsymbol{\eta}_p\right) \sim \text{Rayleigh}(\sigma_n)$. The expected value of the Rayleigh distribution is equal to $\sigma_n\sqrt{\pi/2}$ [146]. To express this analytically, the following equation is used to describe a search space position value or correlation output $S(\tau, f)$ using both the signal and noise content:

$$S(\tau, f) = S_S(\tau, f) + S_N(\tau, f), S_S \in \mathbb{C}, S_N \in \mathbb{C} \tag{3.95}$$

where $S_S(\tau, f)$ is the signal content and $S_N(\tau, f)$ is the noise content at search space position (τ, f).

The value of $S_S(\tau, f)$ is computed using the following equation, similarly to (3.44):

$$S_S(\tau, f) = S_i(\tau, f) + jS_q(\tau, f) \tag{3.96}$$

The expected value of $S_N(\tau, f)$, $E\{S_N(\tau, f)\}$, is expressed as follows:

$$E\{S_N(\tau, f)\} = \frac{\sigma_{\mathrm{IF}}}{2}\sqrt{\frac{\pi}{N_p}} \tag{3.97}$$

These equations express the relationship between signal and noise in the search space or when computing amplitude using SAGE. This relationship is expressed using the true value of τ and f (perfect estimation) to obtain the ratio between the signal magnitude and the expected noise magnitude Λ.

$$\Lambda = 2\sqrt{\frac{(C/N_0)(PIT)}{\pi}} \tag{3.98}$$

Figures 3.4 and 3.5 illustrate (3.98) for $C/N_0 \in [35, 50]$ dB-Hz and a 1 msec integration period. The impact of noise is immediately apparent: the signal magnitude at the signal peak is relatively low relative to the expected magnitude of the noise content. For a serial correlation algorithm like SAGE, even if the algorithm is performing perfectly, the magnitude of noise content will be approximately half the magnitude of signal content when propagation delay and Doppler frequency have been perfectly estimated at $C/N_0 = 35$ dB-Hz. For an iterative algorithm, errors induced by this phenomenon may have a significant impact on the quality of parameter estimates. These figures illustrate why multipath ray parameter estimation processes may benefit from an approach that makes use of parallel correlation.

An algorithm that compares outputs of parallel correlators, such as SDPA, takes advantage of the weak law of large numbers [146]. When the outputs of a large number

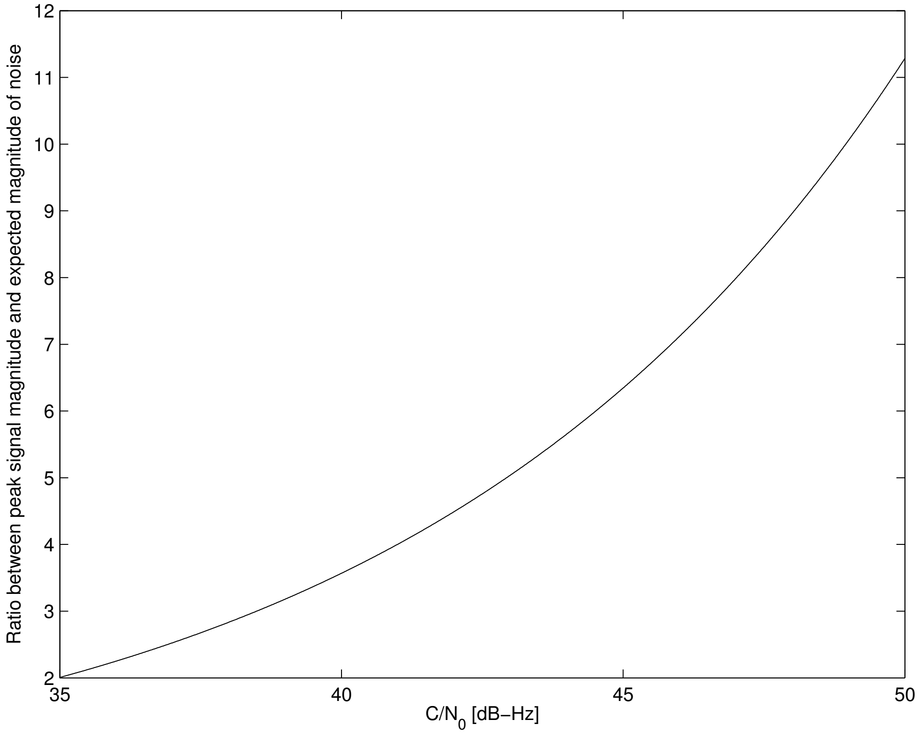

Figure 3.4: Illustration of the ratio between the peak signal magnitude and the expected magnitude of noise content when considering correct propagation delay τ and Doppler frequency f. This ratio is the case for both true signal peak positions in search spaces or as output from the SAGE amplitude computation. This figure is computed using an integration period of 1 msec.

of correlators are considered simultaneously, the sample mean of both the real and imaginary noise content will converge to the true mean of zero, since complex circular AWGN in the frequency domain is independent and identically distributed with zero mean and a fixed variance (thus wide sense stationary). This law of probability justifies exploration of SDPA versus SAGE.

It is important to point out that SAGE makes use of maximum likelihood estimation, so in correlation cases where signal content dominates over noise content, users should not expect SDPA to outperform SAGE. This is, after all, because SDPA is making use of data from regions in the search space where noise content dominates over signal content. As can be seen from Figure 3.5, the ratio of signal magnitude to noise magnitude is significant in the case when $C/N_0 = 50$ dB-Hz, so relative estima-

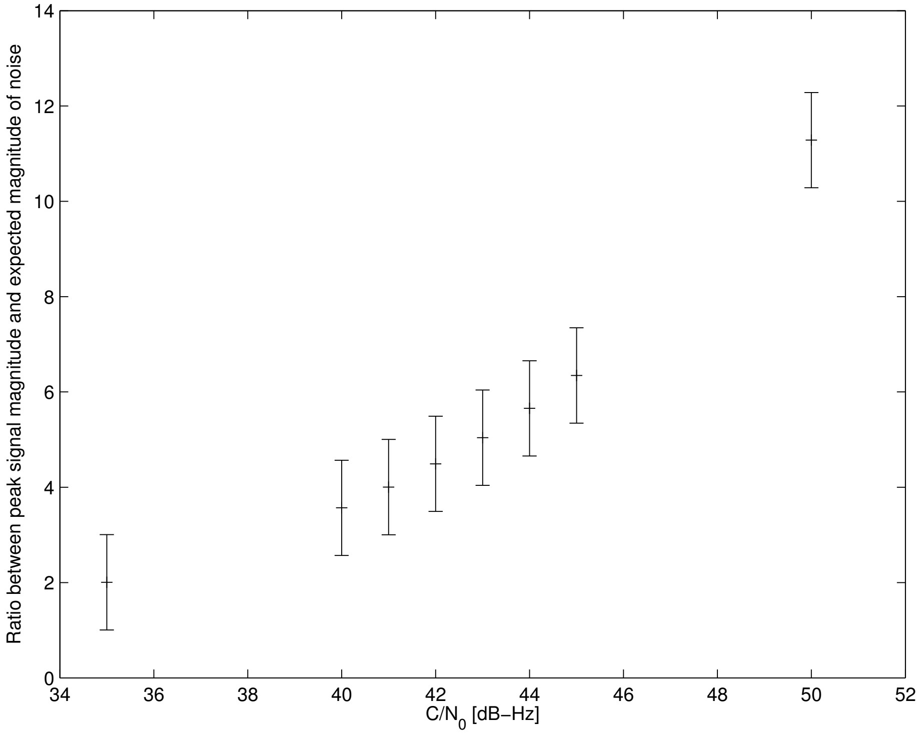

Figure 3.5: Illustration of the ratio between the peak signal magnitude and the expected magnitude of noise content when considering correct propagation delay τ and Doppler frequency f. This ratio is the case for both true signal peak positions in search spaces or as output from the SAGE amplitude computation. This figure is computed using an integration period of 1 msec. For each ray magnitude, the error bar centered by the highest position of the ray magnitude illustrates the ratio of expected deviation brought about because of the noise content for a given value of C/N_0, relative to the peak signal magnitude (the center of the error bar).

tion error will not be as prevalent as that observed when $C/N_0 = 35$ dB-Hz. Since SDPA makes use of much more of the spectrum than SAGE in parameter estimation, there are portions of the search space from which data is extracted even when very little signal content is present, relative to the noise content. This may have the effect of inducing estimate errors not experienced when using SAGE. The solution to this may be to weight correlator magnitudes from areas of greater concentration of signal content more heavily than areas where there is limited signal content.

3.5 Summary

This chapter provides a comprehensive description of SDPA. As part of this discussion, motivation for the consideration of SDPA, versus a maximum likelihood estimator such as SAGE, was presented.

IV. Results and Analysis of Comparisons between SDPA- and SAGE-based Decomposition

4.1 Introduction

SAGE appears to be the state of the art in the decomposition of GNSS signals with the objective of estimating multipath parameters under the following conditions:

- Waveform amplitude, Doppler frequency, propagation delay offset, and carrier phase are explicitly estimated for the direct path and each multipath ray.
- A tracking loop is not required in order to execute the SAGE-based algorithm, in contrast to a technique like MEDLL.

Because of this, SAGE will be the primary point of comparison for SDPA.

As discussed in Section 2.6, the SAGE-based decomposition algorithm can be inserted seamlessly into the SDPA architecture in order to provide a direct, fair comparison between the two methods. In order to demonstrate the effectiveness of SDPA relative to the SAGE-based algorithm, a simulation scenario is constructed. This simulation scenario is designed to replicate what is expected of the receiver hardware and multipath environment in future research, particularly in an urban environment where signals are collected by a TRIGR receiver mounted in an automobile. No claim is made that SDPA will generally outperform SAGE-based decomposition, but this research indicates that consideration of time-frequency domain-based signal processing, such as that used in SDPA, may be appropriate in some scenarios.

This chapter of the dissertation presents results and analysis of performance results between SDPA- and SAGE-based decomposition algorithms. Section 4.2 outlines the simulation scenario that is employed in order to obtain decomposition results for comparison. Section 4.3 provides the decomposition results themselves in a comparative format. Section 4.4 provides analysis of the results obtained in Section 4.3.

4.2 Comparison Simulation Scenario

Simulations are employed to demonstrate the claim that SDPA offers in limited and specific scenarios the ability to better reduce the RMS error between the true signal and the estimate signal when transformed to a time-frequency domain (the search space) versus the SAGE-based decomposition algorithm. To justify this comparison and claim, simulations are used. These simulations consist of the independent generation of 1000 data vectors of a 1 msec integration period for each of the following values of C/N_0: 35, 40, 41, 42, 43, 44, 45, and 50 dB-Hz. Decomposition occurs for each of these data vectors using both SDPA- and SAGE-based decomposition. The objective with these simulations is to demonstrate that SDPA exhibits better estimation performance, in terms of minimization of the RMS error between the estimate search space and the true (no noise) search space for all estimate ensembles obtained when processing a PIT, relative to SAGE-based decomposition.

There are two sets of parameters that compose the specific simulation scenario considered in this research—the receiver parameters for which simulated signals are designed, and the multipath propagation model that has been developed for this research. The receiver parameters are discussed in Section 4.2.1. The simulated signal structure multipath propagation model is discussed in Section 4.2.2. Section 4.2.3 provides the specific parameters associated with the calculation of the search space discussed in Sections 3.3.5 and 3.3.9. Section 4.2.4 discusses assumptions that are made in decomposition and parameterization of simulated signals. These assumptions are made to reduce the computational overhead associated with the simulations without compromising the integrity of the simulation results and findings.

4.2.1 Simulated Receiver Parameters. Future study into the topic area considered in this dissertation will make use of the TRIGR receiver. Therefore, TRIGR parameters are considered in this research. Data sets received using the TRIGR are typically recorded using the following parameters:

- Sampling frequency $f_s = 56.32$ MHz

- Intermediate frequency $f_{\text{IF}} = 13.68$ MHz
- 8-bit quantization
- Real-valued signals

The sampling frequency very quickly becomes computationally burdensome. Therefore, downsampling of signals may make use of the following parameters instead of those listed above. The following parameters are used in this research in place of those listed above.

- Sampling frequency $f_s = 56.32/4$ MHz $= 14.08$ MHz
- Intermediate frequency $f_{\text{IF}} = 13.68/4$ MHz $= 3.42$ MHz

4.2.2 Simulated Signal Structure and Multipath Propagation Model. Signals are simulated strictly using GPS C/A-coded waveforms, which have a chipping frequency of 1.023 MHz. PRN 26 is arbitrarily used throughout all simulations, with no waveforms making use of other PRN sequences present in any generated signals. Therefore, all simulated signals consist strictly of the direct path ray, the multipath rays, and AWGN. The integration period is strictly held to a duration of 1 msec for every simulated signal generated for this research, so simulated data vectors of duration equal to 1 msec will be of length 14080 samples. Though only GPS C/A-coded waveforms are considered in this chapter of the dissertation, the use of SDPA and the simulation methodology discussed in this chapter are both generally applicable to waveforms from other GNSS constellations discussed in Section 2.10 as well.

The integration period of 1 msec is used in this case primarily to expedite simulation processing. However, there is a tradeoff to be made in the generalized selection of the integration period. A larger integration period offers the advantage of a processing gain, where signals will be exhibit an increased magnitude relative to noise because of the use of coherent integration. While this advantage presents an enticement, the disadvantage is that the use of the longer integration period has the effect of reducing the magnitude of high frequency multipath signals relative to lower frequency multipath signals. This loss of high frequency multipath reduces the ability of the algorithm to properly parameterize what is truly presented in the received signal. Given this tradeoff, the research presented in this dissertation made use of a smaller integration period with the intent of proper parameterization of high frequency multipath content.

Literature on GNSS modeling in the urban environment discussed in Section 2.4.2.2 provides the basis for the multipath propagation model used in this research. Papers published by researchers at DLR in particular are heavily leveraged to infer realistic multipath waveform parameter values outlined as follows:

4.2.2.1 Number of ray waveforms. 20 multipath ray waveforms (in addition one direct path waveform) are used for every PIT-sized data vector generated in this research, as inferred by inspection from Figure 14 in [70]. Figure 14 in [70] illustrates the number of echoes arriving at the receiver antenna at the same time during a 15 minute drive through city streets. The figure illustrates the possibility of up to 50 echoes arriving at the receive antenna simultaneously, but the mean value is far fewer. The use of $M_p = 20$ in simulation represents an approximation of the expected value through inspection of the figure, so this number is asserted in every generated instantiation of $\mathbf{r}_p$.

4.2.2.2 Rician K-factor. The Rician K-factor is a ratio of the power in the direct path ray to the power in the sum of multipath rays. A K-factor of one is used in the generation of every PIT-sized data vector. Figure 11 in [70] illustrates a histogram of the Rician K-factor exhibited in multipath rays leading to the inference by inspection that a K-factor of one should be used. This is because the peak value for the K-factor within the histogram is approximately one.

4.2.2.3 Amplitude of ray waveforms. The multipath ray waveform amplitudes values are scaled relative to each other to correspond with models posited by DLR [16]. For multipath ray m, the peak amplitude for the ray amplitude A_m

delayed by τ_m is expressed as follows:

$$A_m(\tau_m) = S_0 e^{-\delta \tau_m} \tag{4.1}$$

where S_0 is the initial value of $A_m(\tau_m)$ when $\tau_m = 0$ and δ is the exponential decay constant. S_0 varies depending upon C/N_0, so a numeric value won't be provided here. In the application of this model in simulations, $\delta = 2.410 \times 10^6$. The value of δ used in simulations conducted for this research has been obtained by analyzing simulation outputs from the DLR land mobile multipath model software associated with [70] and available at [148].

4.2.2.4 Scaling Multipath Rays to Satisfy the K-factor. Multipath rays are scaled relative to the direct path ray to satisfy the ratio prescribed by the K-factor as follows:

$$\mathbf{r}_{p_m} = [\mathbf{r}_{p_m}]_{\text{unscaled}} \sqrt{\frac{P_L}{P_m}} \tag{4.2}$$

where $[\mathbf{r}_{p_m}]_{\text{unscaled}}$ is the version of $\mathbf{r}_{p_m}$ that has been generated before scaling. The direct path signal power P_L (the power of direct path ray $\mathbf{r}_{p_0}$) and the multipath waveform power P_m are expressed as follows:

$$P_L \cong \text{var}([\mathbf{r}_{\mathbf{p0}}]_{\text{unscaled}}) \tag{4.3}$$

and

$$P_m \cong \text{var}\left(\sum_{m=1}^{M} [\mathbf{r}_{p_m}]_{\text{unscaled}}\right) \tag{4.4}$$

This approach to computation of P_L and P_m found in (4.3) and (4.4) makes use of the fact that the power present in a time-domain signal can be approximated by summing the variance of the received signal with the square of the mean of the received signal [3].

4.2.2.5 Propagation delay offset distribution of rays. Each of the multipath rays in each data vector are delayed, relative to the direct path ray, using an exponential distribution. Use of this random distribution is in accordance with the findings reported in [16]. The distribution is expressed as follows:

$$P(\tau_m) = \frac{1}{\mu} e^{-\frac{\tau_m}{\mu}} \tag{4.5}$$

where $\mu = 0.225 \times 10^{-6}$. This value of μ contrasts from that reported in [16] to accommodate the characteristics of the model constructed by DLR and reported in Figure 17 in [70]. μ is adjusted when using the exponential distribution to generate a value of τ_m that approaches but does not exceed a propagation delay of 1.8×10^{-6} sec, relative to the direct path ray.

4.2.2.6 Doppler frequency offset distribution of rays. The Doppler frequency offset for individual rays is uniformly distributed in a range between -60 and 60 Hz around the Doppler frequency of the simulated direct path ray waveform in every PIT. This simulation decision is inferred from Figure 18 in [70]. Figure 18 in [70] illustrates the Doppler frequency of echoes simultaneously present at the receiver antenna as a vehicle travels through city streets. This figure illustrates the possibility of receiving multipath waveforms that exhibit a Doppler frequency shift of as much as 60 Hz. The use of a uniform distribution is inferred by inspection of the figure.

4.2.2.7 Initial carrier phase distribution of rays. The initial carrier phase for individual ray waveforms is uniformly distributed around the unit circle for every ray waveform in every PIT, as discussed in Section 2.4.2.2 and per [16].

4.2.3 SDPA Search Space Parameters. In the construction of the search spaces described in Sections 3.3.5 and 3.3.9, there are several parameters for which values are chosen for this research: $\tau_{window} = 2T_c$, $f_{window} = 10$ kHz, $\Delta\tau_{input} = T_s$, and

$\Delta f_{input} = 10$ Hz. τ_{window} is chosen to be equal to $2T_c$ to ensure that multipath that is within T_c of the direct path signal, in terms of propagation delay, will be present in the search space. The area of interest in this research, in terms of the multipath signal relative to the direct path signal, is of a propagation delay within T_c of the direct path signal. The frequency window size f_{window} of 10 kHz may be wider than what would be expected, but is used to account for large Doppler shifts and clock drifts, and is used in accordance with [67]. $\Delta\tau_{input}$ is chosen to be equal to T_s because this value provides the highest possible resolution (a smaller number does not contribute to the resolution, as slices in the search space will simply be duplicated and a larger number will cost the algorithm resolution). Δf_{input} is chosen to be equal to 10 Hz arbitrarily with the awareness that relatively tight grid spacing offers better initial frequency estimates, as described in Section 3.3.7.

4.2.4 Assumptions to Reduce the Computational Burden in Simulation. For the simulations performed for the research outlined in this dissertation, there is an ensemble index beyond which the error between the estimate search space and the "true" (noise-free) search space will not be further reduced through more processing. Continuing decomposition of the current integration period would beyond this point is needless and time-consuming. To reduce the computational burden in simulation, offline simulations are performed to determine the ensemble index for each value of C/N_0 at which decomposition of the received data for the current integration period can be halted without impacting the results.

For $C/N_0 = 45$ dB-Hz and a 1 msec integration period, 1000 independent trial data vectors are generated and decomposed using both SDPA and SAGE. Decomposition of this generated data led to the finding that the error between the estimate and true search spaces is minimized within the first 17 ensembles. However, with the exception of a single decomposition trial, the error between the estimate and true search spaces is minimized within the first 10 ensembles. Therefore, it is assumed that processing 12 ensembles will generally suffice in comparing SDPA and SAGE.

Furthermore, since the number of ensembles required to reach the error minimum is reduced as C/N_0 is reduced, this assumption is made for simulations using values of C/N_0 less than 45 dB-Hz.

For $C/N_0 = 50$ dB-Hz and a 1 msec integration period, 1000 independent trial data vectors are generated and decomposed using both SDPA and SAGE. Decomposition of this generated data led to the finding that the error between the estimate and true search spaces is minimized within the first 17 ensembles. Therefore, it is assumed this is the case generally. Simulations with $C/N_0 = 50$ dB-Hz are halted after processing 17 ensembles.

4.3 Comparison of Results between SDPA- and SAGE-based Decomposition Techniques

Given the simulation scenario described in the previous section (to include the assumptions made regarding the maximum number of ensembles required to obtain the error minimum), 1000 independent trials are generated at each of the following values of C/N_0: 35, 40, 41, 42, 43, 44, 45, and 50 dB-Hz. SDPA and SAGE are independently employed in the decomposition of a data vector of duration equal to 1 msec. There are three performance metrics that are considered directly in the comparison of SDPA and SAGE in the decomposition and parameterization of data. The first metric is the proportionality of error minimization between SDPA and SAGE. The second metric is the mean error minimum obtained by SDPA or SAGE. The third is the median error minimum obtained by SDPA or SAGE. To obtain performance measurements for each of these three metrics, trials are executed where SDPA and SAGE are both independently employed to decompose the data associated with the trial.

Figure 4.1 is a bar chart illustrating the results associated with the first metric, the proportionality of error minimization between SDPA and SAGE. There are three possible outcomes when each of the 1000 trials are evaluated. The first possibility is the "invalid" result, where either SDPA or SAGE are unsuccessful in decomposing the region of the search space where direct or multipath content is present. This is determined by whether or not parameter outputs obtained through decomposition (using every ensemble processed for the trial) place a ray waveform at positions both less than and greater than the peak of the true search space, in terms of both Doppler frequency and path length offset. In other words, if the peak of the true search space isn't "bookended" in both Doppler frequency and path length offset by rays output from the decomposition engine using both SDPA and SAGE, the decomposition trial is considered to be invalid. To compare valid versus invalid results in this context is really more a test of the acquisition performance of the decomposition engine, akin to how GNSS signals are acquired in conventional receivers. However, reporting of these results illustrate how realistic it is to attempt to decompose data at various values of C/N_0, therefore the results are retained and reported. If decomposition is considered to be valid, then SDPA and SAGE are compared to determine which of the two algorithms succeed in reaching the lower error value when comparing estimate and true search spaces. If SDPA reaches the lower of the two error minima obtained through use of the two algorithms, SDPA is credited with success, otherwise SAGE is credited with success. It should by noted that these error minima are not necessarily the global minimum that could possibly be achieved with an arbitrary estimation algorithm, but only the best achieved with these two algorithms. Figure 4.1 illustrates the finding that SDPA succeeds in reaches the error minimum more often than SAGE for $C/N_0 \leq 40$ dB-Hz.

Figure 4.2 illustrates a comparison of the mean error minimum obtained using both SDPA and SAGE. The error minimum for valid trials for either SDPA or SAGE are averaged together and reported in this figure, as well as the standard deviation associated with the error minimum values. Note that these error figures are normalized. Each of the error minimum values for the valid trials for both SDPA and SAGE are divided by the mean error minimum obtained using SAGE. Because these results are normalized using SAGE error minimum results, the error minimum for SAGE using each value of C/N_0 is equal to one. This normalization enables a relativistic

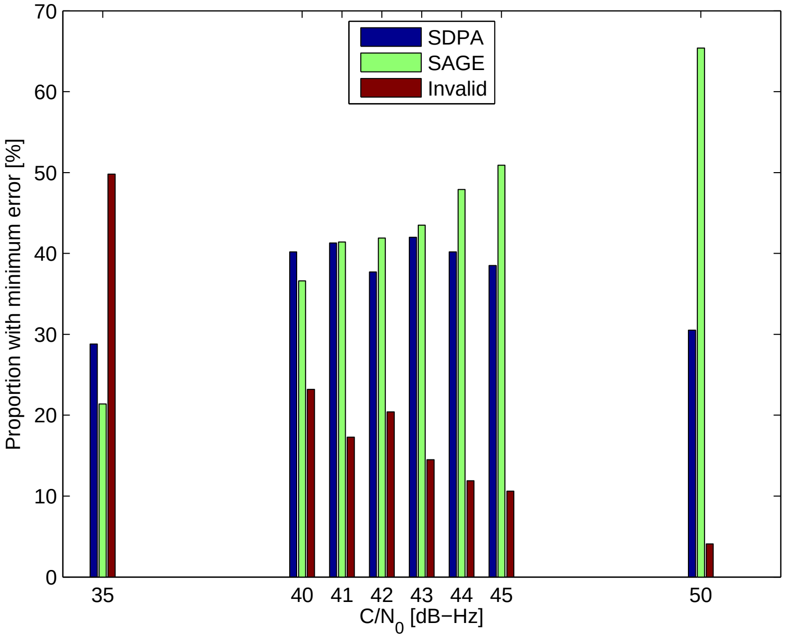

Figure 4.1: Illustration of proportionality of error minimization for $C/N_0 \in \{35, 40, 41, 42, 43, 44, 45, 50\}$ dB-Hz. The proportionality of instances for each value of C/N_0 where SDPA, versus SAGE, obtained the minimum error between the estimate and true search spaces is displayed in blue. The proportionality of instances for each value of C/N_0 where SAGE, versus SDPA, obtained the minimum error between the estimate and true search spaces is displayed in green. The proportionality of instances for each value of C/N_0 where decomposition results were invalid is displayed in red. This bar chart illustrates that SDPA succeeds more frequently in reaching the error minimum, versus SAGE, at $C/N_0 \leq 40$ dB-Hz.

comparison between SDPA and SAGE. Although standard deviations are provided with error bars in this figure, it needs to be reported that the error minima are not normally distributed, but actually appear to be Rice distributed in nature. Therefore, the further use of standard deviation beyond this finding may not be advisable. This figure illustrates that SDPA succeeds in reaching the lower mean error minimum, versus SAGE, at C/N_0 equal to 35 and 40 dB-Hz. Of note is the finding that the standard deviation for error minimum value is widely varying, so decomposition of one trial integration period may have a very different error minimum from the next.

Figure 4.3 illustrates a comparison of the median error minima obtained using both SDPA and SAGE. The first and third quartiles are annotated in this figure as

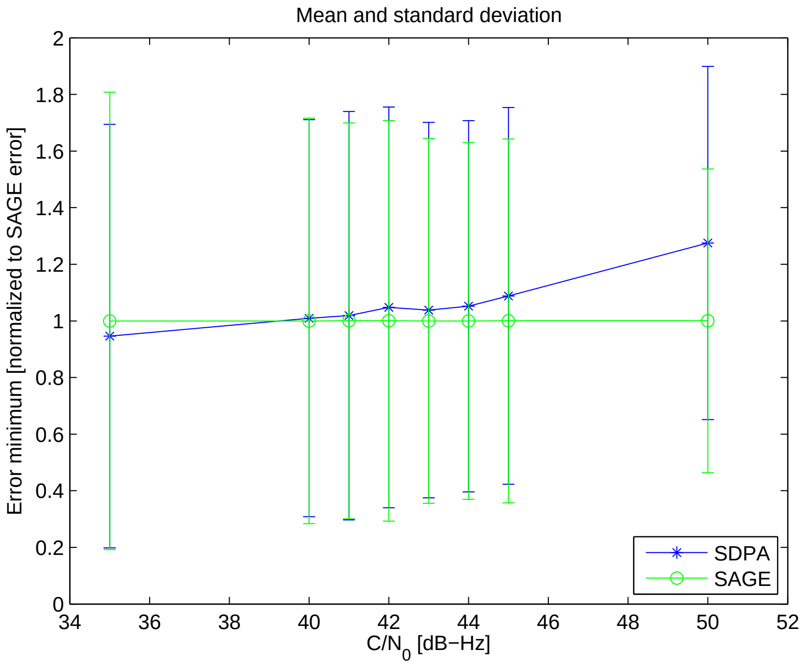

Figure 4.2: Comparison of the mean error minimum obtained using both SDPA and SAGE for $C/N_0 \in \{35, 40, 41, 42, 43, 44, 45, 50\}$ dB-Hz. The mean error minimum for SDPA is presented using blue asterisks. The mean error minimum for SAGE is presented using green circles. The error bar illustrates the standard deviations for both SDPA and SAGE. The blue error bar illustrates the standard deviation of the error minima using SDPA, whereas the SAGE error bar is illustrated in green. This figure illustrates that SDPA succeeds in reaching the lower mean error minimum, versus SAGE, at C/N_0 equal to 35 and 40 dB-Hz. The large standard deviation for both SDPA and SAGE indicates that the error minimum varies widely from one integration period to the next.

well for both SDPA and SAGE for each value of C/N_0. The normalization associated with the results illustrated in Figure 4.2 is retained for these results as well. This figure illustrates that SDPA succeeds in reaching the lower median error minimum, versus SAGE, at $C/N_0 \leq 40$ dB-Hz. Of note is the finding that the range between the first and third quartiles in error minimum values is widely varying, so decomposition of one trial integration period may have a very different error minimum from the next. Further indication that SDPA performs better than SAGE at $C/N_0 \leq 40$ dB-Hz is the shift in which algorithm has the more desirable third quartile (upper) error bar.

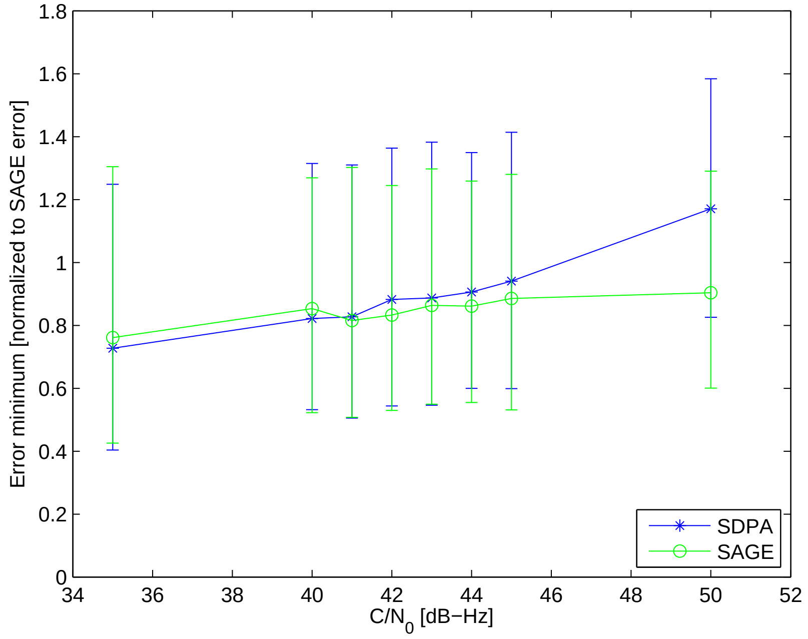

Figure 4.3: Comparison of the median error minima obtained using both SDPA and SAGE for $C/N_0 \in \{35, 40, 41, 42, 43, 44, 45, 50\}$ dB-Hz. The median error minimum for SDPA is presented using blue asterisks. The median error minimum for SAGE is presented using green circles. The error bar illustrates the first and third quartile positions for both SDPA and SAGE. The blue error bar illustrates the quartile positions of the error minima using SDPA, whereas the SAGE error bar is illustrated in green.

Tables 4.1 and 4.2 provide statistics for SDPA and SAGE, respectively, on the probability that a given decomposition ensemble $\varepsilon_p \in \{0, 1, \ldots, 16\}$ yielded the error minimum between the estimate and true search spaces. Comparison between these two tables provide a couple of insights. First, the assumptions outlined in Section 4.2.4 on fixing the maximum number of ensembles permitted for a given integration period are justified, given the sparsity of error minimums at higher values of ε_p. Second, use of SDPA typically yields a greater number of decomposition ensembles than SAGE, so a greater number of rays will typically compose the error minimum for a given value of C/N_0 if SDPA is used, versus SAGE. This insight is not globally the case, but is typical. Cells within the table may be rounded such that the sum of cells in a column varies slightly from one.

Table 4.1: Probability that SDPA decomposition ensemble index ε_p yields the error minimum between the estimate and true search spaces for a given value of C/N_0.

	C/N_0							
ε_p	35	40	41	42	43	44	45	50
0	0.958	0.915	0.924	0.884	0.843	0.809	0.780	0.444
1	0.026	0.064	0.053	0.082	0.112	0.132	0.138	0.225
2	0.016	0.013	0.013	0.020	0.028	0.043	0.049	0.172
3	0	0.007	0.007	0.008	0.009	0.008	0.024	0.077
4	0	0.001	0.002	0.004	0.002	0.006	0.003	0.027
5	0	0	0	0.003	0.002	0.002	0.003	0.018
6	0	0	0	0	0	0	0.002	0.008
7	0	0	0	0	0	0	0	0.007
8	0	0	0	0	0	0	0.001	0.002
9	0	0	0	0	0.002	0	0	0.005
10	0	0	0	0	0	0	0	0.001
11	0	0	0	0	0	0	0	0.005
12	0	0	0	0	0	0	0	0.003
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0.003
16	0	0	0	0	0	0	0	0.001

Table 4.2: Probability that SAGE decomposition ensemble index ε_p yields the error minimum between the estimate and true search spaces for a given value of C/N_0.

	C/N_0							
ε_p	35	40	41	42	43	44	45	50
0	0.968	0.944	0.929	0.895	0.862	0.817	0.772	0.393
1	0.030	0.055	0.069	0.101	0.130	0.162	0.211	0.513
2	0.002	0.001	0.002	0.004	0.008	0.019	0.016	0.081
3	0	0	0	0.001	0	0.001	0.001	0.008
4	0	0	0	0	0	0	0	0.003
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0.001
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0

4.4 Analysis of Decomposition Results

These results clearly indicate that at higher values of C/N_0, the SAGE-based algorithm provides superior performance in the estimation of the search space for received GNSS data. However, as noise becomes more dominant, SDPA is indicated to be the more promising choice for decomposing received data. Section 3.4.2 discusses the motivation for developing SDPA and comparing it to SAGE. SAGE is believed to be the state-of-the-art in the decomposition of a data vector of duration equal to the integration period. As discussed in Section 3.4.2 in the context of estimating ray waveform amplitude values, the serial correlation nature of SAGE presents the risk of noise content being weighted more heavily relative to signal content. This weighting of noise may lead to outsize or undersized amplitude estimates for one ray waveform, which impacts the simultaneous estimation of other rays in the search space. This impact leads to a loss of opportunity in accurately estimating ray waveforms. The motivation for comparing SDPA and SAGE outlined in Section 3.4.2 is revealed to be founded.

4.5 Summary

This chapter outlines the results of comparisons between SDPA and SAGE using simulated data. These results clearly indicate that at lower values of C/N_0 (particularly at values less than or equal to 40 dB-Hz when using a 1 msec integration period), SDPA outperforms SAGE in the decomposition of received data.

V. Stopping Criteria Methodology, Results, and Analysis

5.1 Introduction

The stopping criteria is required in order to dictate to a decomposition algorithm that processing of an integration period is to be halted. This is in lieu of a technique specifically designed to estimate the number of received multipath rays present in an integration period. The use of a stopping criteria follows the trend in literature (as discussed in Section 2.8.4) of making use of the residual to determine whether the algorithm should be halted.

This chapter of the dissertation presents the methodology, results and analysis of the stopping criteria employed to halt decomposition and parameterization of PIT p for both SDPA and SAGE. Section 5.2 discusses both the methodology and the rationale behind the methodology used to establish the stopping criteria. Section 5.3 outlines the simulation scenario that is employed to obtain results required to establish the stopping criteria. Section 5.4 provides the stopping criteria results themselves. Section 5.5 provides analysis of the results obtained in Section 5.4.

5.2 Stopping Criteria Methodology

The procedure for establishing the threshold Υ to be used for the stopping criteria is outlined in this section of the document. The stopping criteria threshold Υ is discussed in Section 3.3.13 and used specifically to provide the decision threshold by which the stopping criteria statistic F_{p_ε} is evaluated, as expressed in (3.91). This section provides a detailed explanation of the process by which Υ is obtained, and extends the explanation provided previously in [142].

The objective in establishing the stopping criteria threshold is to define the most likely value of F_{p_ε} for which the RMS error $\mathcal{E}_{p_\varepsilon}$ is minimized between the estimated signal content within a search space $\hat{\mathbf{R}}_{p_\varepsilon}$ and the "true" signal content within a search space $\mathcal{S}_p$. This value serves as the threshold Υ. If F_{p_ε} is greater than Υ, then it is asserted that further iterations of the decomposition process will reduce $\mathcal{E}_{p_\varepsilon}$. If

F_{p_ε} is equal to or less than Υ, then it is asserted that no further iterations of the decomposition process will reduce $\mathcal{E}_{p_\varepsilon}$.

There is a distinct difference between $\mathcal{E}_{p_\varepsilon}$ and E_{p_ε}, in that the decomposition algorithm absolutely has the potential to output ensembles containing ray waveforms that have parameterized noise content within $\mathbf{R}_p$. If noise content is parameterized, and the ray waveform associated with this noise parameterization is included in $\hat{\mathbf{R}}_{p_\varepsilon}$, then E_{p_ε} will generally be reduced, indicating a closing of the gap between the received and the estimate search grids. $\mathcal{E}_{p_\varepsilon}$, on the other hand, will generally increase, indicating a widening of the gap between the estimate and truth search grids. The algorithm is intended to decompose and parameterize as much signal content contained within the received search grid as possible. Simultaneously, the algorithm is intended to decompose and parameterize as little noise content contained within the received search grid as possible. The intent of embedding a stopping criteria with a threshold Υ within the decomposition process is to provide a means by which the algorithm manages this tension between competing goals. If there is no objection to parameterizing noise content, then the algorithm may be permitted to decompose indefinitely. There are, however, two problems with this approach. The first problem is that noise parameters provide no value to SDPA and the presence of noise parameters may disrupt efforts to track multipath parameters over time. The second problem with indefinite decomposition is that processing time is wasted on the decomposition and parameterization of noise content that does not provide any additional illumination to the algorithm user regarding the characteristics of the local environment surrounding the GNSS receiver antenna. In short, decomposing and parameterizing noise content wastes time and may create confusion for a notional automated multipath parameter tracking algorithm that would make use of SDPA direct path and multipath ray parameter estimates, so a stopping criteria is established in order to maximize the decomposition and parameterization of signal content while simultaneously minimizing the decomposition and parameterization of noise content.

$\mathcal{S}_p$ is defined as follows, making use of (3.31):

$$\mathcal{S}_p = \mathrm{grid}\left(\sum_{m=0}^{M_p} \mathbf{r}_{p_m}, \tau_{window}, f_{window}, \Delta\tau_{input}, \Delta f_{input}\right) \tag{5.1}$$

To compute the RMS error $\mathcal{E}_{p_\varepsilon}$ between $\hat{\mathbf{R}}_{p_\varepsilon}$ and $\mathcal{S}_p$, the following equation is used:

$$\mathcal{E}_{p_\varepsilon} = \sqrt{\frac{1}{N_\tau N_f} \sum_{y=0}^{N_\tau - 1} \sum_{z=0}^{N_f - 1} \left| \mathcal{S}_p - \hat{\mathbf{R}}_{p_\varepsilon} \right|^2} \tag{5.2}$$

Before the stopping criteria was included in the SDPA, the algorithm would continue processing until E_{p_ε} could not be further reduced. $\mathcal{E}_{p_\varepsilon}$, on the other hand, typically reaches a minimum after an earlier iteration than the final iteration available for p. The objective in defining and deploying the threshold Υ is to direct the algorithm to halt processing when the minimum value of $\mathcal{E}_{p_\varepsilon}$ is reached. Because the SDPA is designed to decompose and parameterize generic signals described by (3.20) and obtained from undefined environments where outside information (which could be used to provide articulation of the environment from which a received signal originates) is not permitted to be included in the algorithm, a formal derivation of the optimum threshold Υ is not considered to be practical for this research. If the received waveform $\mathbf{r}_p$ described in (3.20) is characterized prior to SDPA processing, and information could be provided to the algorithm to provide further definition to the structure of $\mathbf{r}_p$, a formal derivation would be much more tractable.

Υ is determined by relating the true iteration after which to halt decomposition with the mean iteration from simulation after which to halt decomposition [142]. More specifically, Υ is defined as follows:

$$\Upsilon = \arg\min_{\psi} \left(\sqrt{\frac{1}{\Sigma_P} \sum_{\substack{\mu=0 \\ P_\mu > 0}}^{\mu_{\max}} \left(\bar{\varepsilon}_{\nu|\mu,\psi} - \mu\right)^2} \right) \tag{5.3}$$

where $\psi \in [0,1]$, P_μ is the number of integration periods considered for which the ensemble index ε that minimizes $\mathcal{E}_{p_\varepsilon}$ for PIT p is equal to μ, $\mu_{\max}$ is the largest ensemble index ε for which $\mathcal{E}_{p_\varepsilon}$ is minimized among all integration periods considered, and Σ_P and $\bar{\varepsilon}_{\nu|\mu,\psi}$ are defined using the following two equations, respectively:

$$\Sigma_P = \sum_{\substack{\mu=0 \\ P_\mu > 0}}^{\mu_{\max}} 1 \tag{5.4}$$

$$\bar{\varepsilon}_{\nu|\mu,\psi} = \frac{1}{P_\mu} \sum_{\substack{p=0 \\ \varepsilon_{p_\mu}=\mu}}^{p_{\max}} \left(\varepsilon_{p_\psi}\right) \tag{5.5}$$

where ε_{p_μ} is the ensemble index for PIT p in which $\mathcal{E}_{p_\varepsilon}$ is minimized and ε_{p_ψ} is the ensemble index for PIT p after which decomposition processing would halt given trial threshold ψ.

The equations above provide a rigorous mathematical explanation for how the stopping criteria threshold is obtained, but the equations themselves don't communicate simply how the stopping criteria threshold is determined. In words, the idea behind the stopping criteria again is to relate the average ensemble index after which decomposition would halt given trial threshold ψ, $\bar{\varepsilon}_{\nu|\mu,\psi}$, to the ensemble index after which decomposition should actually halt (the ensemble index for PIT p after which $\mathcal{E}_{p_\varepsilon}$ is minimized). $\bar{\varepsilon}_{\nu|\mu,\psi}$ is expected to increase linearly with unity slope as μ is incremented. By taking this approach to determining Υ, arbitrary constraints that would be placed in the decomposition process through use of a ROC curve or another detection theoretic technique found in literature are not placed on the SDPA process. The value of ψ for which the RMS error is minimized between μ and $\bar{\varepsilon}_{\nu|\mu,\psi}$ for every value of μ where a value of $\bar{\varepsilon}_{\nu|\mu,\psi}$ is asserted as Υ. With noise-free data, the mean decomposition iteration after which to halt when applying Υ should equal the true iteration after which to halt [142]. In this research Υ is determined experimentally through simulation.

5.2.1 ROC Curve Generation. As discussed in Section 2.8.2, correct detection occurs in cases when SDPA algorithm iteration progresses the algorithm towards reaching the error minimum between the truth and the estimate or iteration has progressed to the point when the error minimum is reached with the current iteration. A false alarm occurs in iteration when the error minimum has already been reached and further iteration is not necessary. Therefore, the probability of correct detection P_d for threshold $\psi \in [0, 1]$ is computed as follows:

$$P_d(\psi) = \frac{\sum_{p=0}^{p_{\max}} (\Sigma_1)}{\sum_{p=0}^{p_{\max}} \left(\varepsilon_{p_\mu}\right)} \tag{5.6}$$

where

$$\Sigma_1 = \sum_{\substack{\varepsilon_p = 0 \\ F_{p_\varepsilon} \geq \psi}}^{\varepsilon_{p_\mu} - 1} 1 \tag{5.7}$$

The probability of false alarm P_{fa} for threshold ψ is computed as follows:

$$P_{fa}(\psi) = \frac{\sum_{p=0}^{p_{\max}} (\Sigma_0)}{\sum_{p=0}^{p_{\max}} \left((\varepsilon_p)_{final} - \varepsilon_{p_\mu} + 1 \right)} \tag{5.8}$$

where $(\varepsilon_p)_{final}$ is the last ensemble index generated during the decomposition process when $\Upsilon = 0$ and

$$\Sigma_0 = \sum_{\substack{\varepsilon_p = \varepsilon_{p_\mu} \\ F_{p_\varepsilon} \geq \psi}}^{(\varepsilon_p)_{final}} 1 \tag{5.9}$$

5.3 Stopping Criteria Simulation Scenario

Simulations are employed to develop a stopping criteria to be used within a decomposition algorithm. Simulations consist of the independent generation of 1000 instances of a 1 msec integration period for each of the following values of C/N_0: 40, 45, and 50 dB-Hz. This range of values for C/N_0 provides evaluation of C/N_0 in equal proportions between values of C/N_0. This proportionality should provide threshold information that is applicable to processing received data recorded in a challenged urban environment. The same assumptions employed to reduce computational burden that are discussed in Section 4.2.4 are applied for stopping criteria simulations as well. In fact, the same data sets for C/N_0 equal to 40, 45, and 50 dB-Hz are used. Decomposition occurs for each of these data vectors with the stopping criteria threshold $\Upsilon = 0$, and the results of decomposition are concatenated. This concatenation yields a waveform parameter data set consisting of decomposition outputs for 3000 integration periods. This waveform parameter data set is then evaluated using the operations outlined in Section 5.2 to obtain an appropriate value of Υ. The receiver parameters are discussed in Section 4.2.1. The simulated signal structure multipath propagation model is discussed in Section 4.2.2. The parameters used to construct the search spaces found in Sections 3.3.5 and 3.3.9 are found in Section 4.2.3.

5.4 SDPA- and SAGE-based Decomposition Stopping Criteria Simulation Results

Section 5.4.1 discusses and illustrates the results leading to the determination of the optimal stopping criteria threshold Υ for SDPA-based processing. Section 5.4.2 discusses and illustrates the results leading to Υ for SAGE-based processing. The definition of optimality in the context of the figures provided in this section of the dissertation is provided explicitly by (5.3).

5.4.1 SDPA. Figure 5.1 illustrates the application of (5.3) in order to determine the optimal threshold value $\Upsilon = 0.985$. Figure 5.2 illustrates the scenario by which the RMS error associated with the optimal threshold value Υ illustrated in Figure 5.1 is obtained. Figure 5.3 illustrates a suboptimal scenario where the threshold is 0.01 less than the optimal scenario illustrated in Figure 5.2. This scenario is a permissive scenario, where decomposition is allowed to continue beyond the point where the optimal number of ensembles are obtained. Figure 5.4 illustrates a suboptimal scenario where the threshold is 0.01 greater than the optimal scenario illustrated in

Figure 5.2. This scenario is a restrictive scenario, where decomposition is not allowed to continue to the point where the optimal number of ensembles are obtained. Figure 5.5 illustrates the ROC curve generated using (5.6) and (5.8).

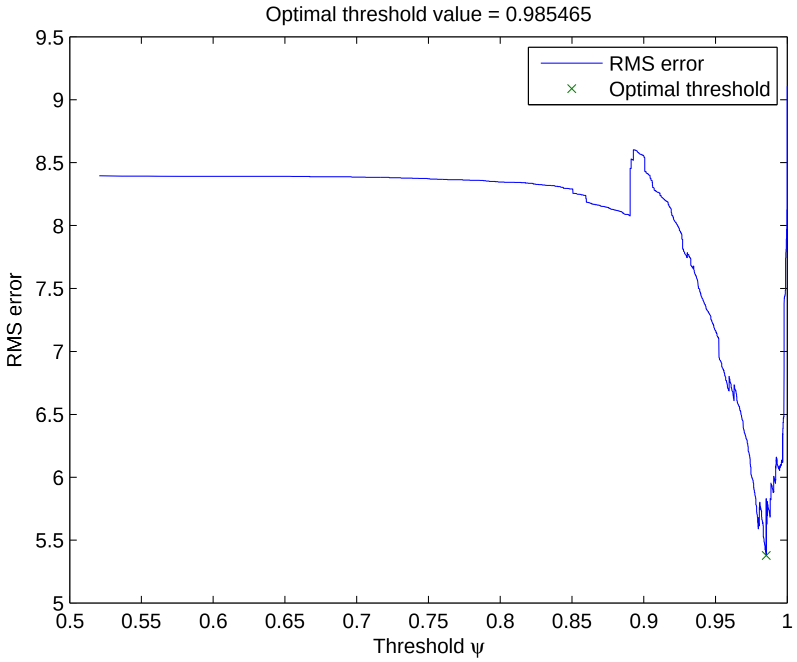

Figure 5.1: Illustration of the application of (5.3) in order to determine the optimal threshold value $\Upsilon = 0.985465$ for SDPA. The blue line represents the RMS error between the true iteration to stop (μ) and the mean stopping iteration based on the threshold value ($\bar{\varepsilon}_{\nu|\mu,\psi}$) (among all values of $\mu \in \{0, 1, \ldots, \mu_{\max}\}$) for a given trial threshold value ψ. The green asterisk represents the position where the minimization argument in (5.3) is satisfied, so $\psi = \Upsilon$ for this position. Note that the value of Υ can be determined definitively, given this resulting illustration from simulation.

5.4.2 SAGE. Figure 5.6 illustrates the application of (5.3) in order to determine the optimal threshold value $\Upsilon = 0.984063$. Figure 5.7 illustrates the scenario by which the RMS error associated with the optimal threshold value Υ illustrated in Figure 5.6 is obtained. Figure 5.8 illustrates a suboptimal scenario where the threshold is 0.01 less than the optimal scenario illustrated in Figure 5.7. This scenario is a permissive scenario, where decomposition is allowed to continue beyond the point where the optimal number of ensembles are obtained. Figure 5.9 illustrates a suboptimal scenario where the threshold is 0.01 greater than the optimal scenario illustrated in

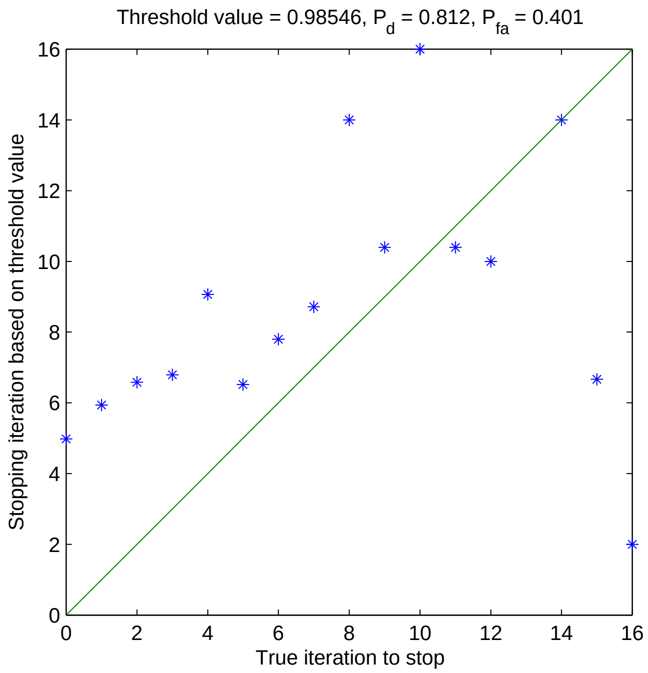

Figure 5.2: Illustration of the RMS error scenario by which the RMS error associated with the optimal threshold value $\Upsilon = 0.985465$ is obtained for SDPA. The blue asterisks are the mean stopping iteration values ($\bar{\varepsilon}_{\nu|\mu,\psi}$) given the true stopping iteration value μ. The green line illustrates the ideal relationship between the true iteration to stop and the mean stopping iteration based on the threshold value ψ. The closer the slope to unity of a line fit obtained from the values of $\bar{\varepsilon}_{\nu|\mu,\psi}$, the better the threshold, per (5.3). Note that the optimal threshold yields a probability of detection $P_d = 0.812$ and a probability of false alarm $P_{fa} = 0.401$.

Figure 5.7. This scenario is a restrictive scenario, where decomposition is not allowed to continue to the point where the optimal number of ensembles are obtained. Figure 5.10 illustrates the ROC curve generated using (5.6) and (5.8).

5.5 Analysis of Decomposition Stopping Criteria Results

Figures 5.1 and 5.6 illustrate how the optimal threshold value Υ for both SDPA and SAGE are obtained. Both of these figures provide the satisfying result that the descent into the error minimum is steep, indicating that the algorithm used to obtain the stopping criteria threshold provides a sharp delineation between values of ψ above Υ versus values of ψ that are below Υ. The threshold values differ significantly

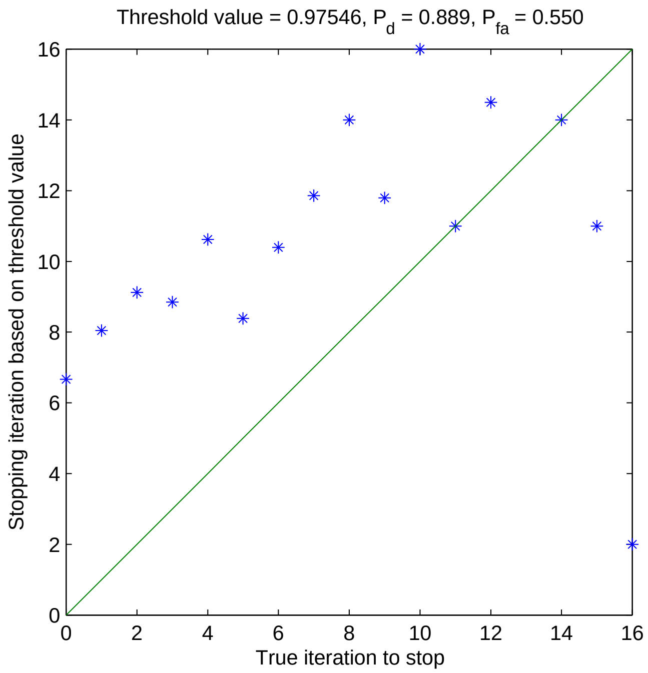

Figure 5.3: Illustration of the RMS error scenario by which the RMS error associated with a permissive threshold value of $\Upsilon - 0.01$ is obtained for SDPA. The blue asterisks are the mean stopping iteration values ($\bar{\varepsilon}_{\nu|\mu,\psi}$) given the true stopping iteration value μ. The green line illustrates the ideal relationship between the true iteration to stop and the mean stopping iteration based on the threshold value ψ. The closer the slope to unity of a line fit obtained from the values of $\bar{\varepsilon}_{\nu|\mu,\psi}$, the better the threshold, per (5.3). Note that this permissive threshold yields a probability of detection $P_d = 0.889$ and a probability of false alarm $P_{fa} = 0.550$.

from those reported in [142], but the simulated receiver parameters contrast significantly between [142] and the simulations presented in this document as well, so this discrepancy is not a cause for alarm.

Figures 5.2 and 5.7 contrast significantly, in that the SDPA error scenario in Figure 5.2 appears to be similar in nature to results in [142]. Figure 5.7, on the other hand, does not even remotely approach the linear trajectory of data points as μ increases. This suggests that the stopping criteria methodology may not be as suitable for SAGE as for SDPA. In considering the results in Table 4.2, there just simply may not be enough instances where SAGE-based decomposition has valid results beyond

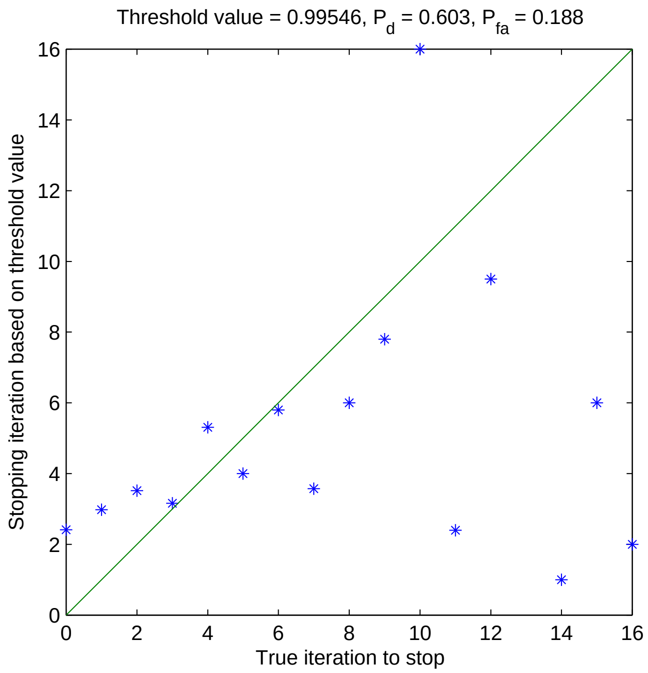

Figure 5.4: Illustration of the RMS error scenario by which the RMS error associated with a restrictive threshold value of $\Upsilon + 0.01$ is obtained for SDPA. The blue asterisks are the mean stopping iteration values ($\bar{\varepsilon}_{\nu|\mu,\psi}$) given the true stopping iteration value μ. The green line illustrates the ideal relationship between the true iteration to stop and the mean stopping iteration based on the threshold value ψ. The closer the slope to unity of a line fit obtained from the values of $\bar{\varepsilon}_{\nu|\mu,\psi}$, the better the threshold, per (5.3). Note that this restrictive threshold yields a probability of detection $P_d = 0.603$ and a probability of false alarm $P_{fa} = 0.188$.

the first few ensembles for decomposition of a given integration period, therefore data obtained for points where μ is relatively high may be too noisy to reasonably use.

Figures 5.3 and 5.4 illustrate the use of permissive and restrictive thresholds, respectively, for SDPA. These figures are provided for comparison with Figure 5.2. Judging from inspection of the three figures, the optimal threshold does indeed appear to originate from the figure that best illustrates the expected unity slope in data point values. At $\mu > 14$, there does appear to be some noise in the data point values in Figure 5.2. This noise, however, does not appear to significantly impact the outcome. The general preference is to obtain more ensembles per integration period, rather than fewer, in order to have more data that can then be further processed at a

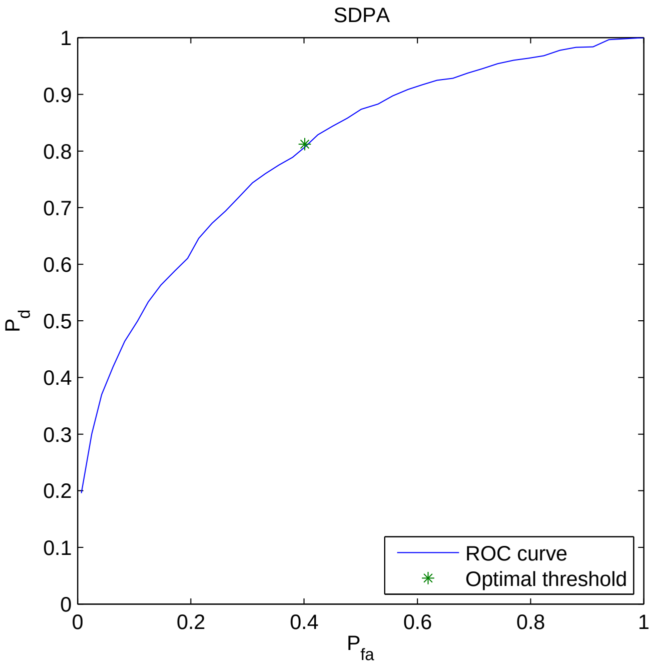

Figure 5.5: Receiver Operating Characteristic (ROC) curve computed using (5.6) and (5.8) for decomposition trials using SDPA. The blue curve is the ROC curve itself. The green asterisk indicates the position on the ROC curve associated with the optimal threshold value $\Upsilon = 0.985465$. The optimal threshold yields a probability of detection $P_d = 0.812$ and a probability of false alarm $P_{fa} = 0.401$. The ROC curve computation makes use of $(\varepsilon_p)_{final}$, which is bound by the the assumptions discussed in Section 4.2.4. As $(\varepsilon_p)_{final}$ increases, P_{fa} is reduced, so constraining $(\varepsilon_p)_{final}$ has the effect of distorting the true ROC curve, as computed using (5.6) and (5.8). Because of this distortion, the ROC curve illustrated in this figure should not be treated as conclusive. The purpose of including the ROC curve is to provide a comparative instrument between SDPA and SAGE.

later time. Therefore, the positioning of data points above the unity slope line is acceptable. The optimal threshold probability of correct detection of 0.812 is 7.7% less than the permissive threshold probability of correct detection in Figure 5.3, while the optimal threshold probability of false alarm is 14.9% less than the permissive threshold probability of false alarm. This indicates that the optimum threshold is indeed preferred over the permissive threshold. The optimal threshold probability of correct detection is 20.9% more than the restrictive threshold probability of correct detection in Figure 5.4, while the optimal threshold probability of false alarm is 21.3% less than the restrictive threshold probability of false alarm. While this indicates that

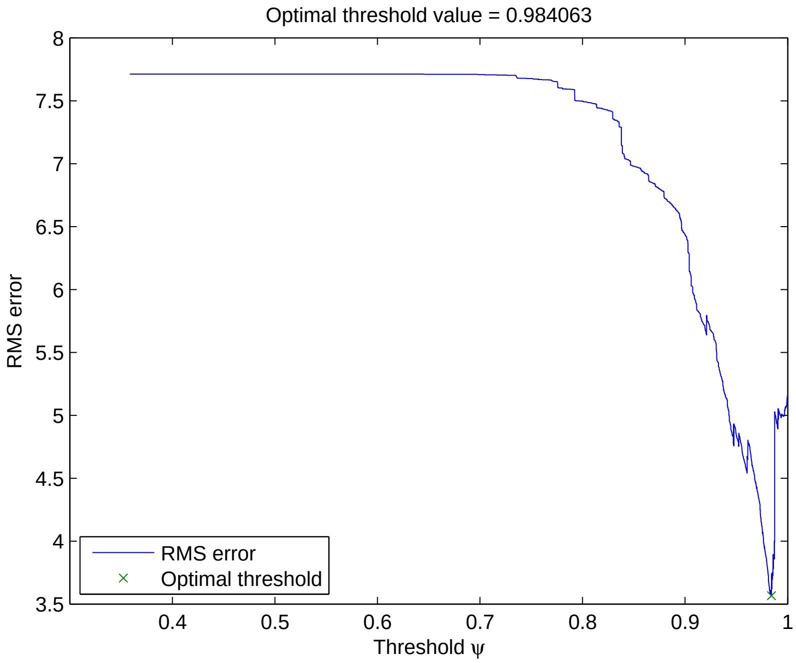

Figure 5.6: Illustration of the application of (5.3) in order to determine the optimal threshold value $\Upsilon = 0.984063$ for SAGE. The blue line represents the RMS error between the true iteration to stop μ and the mean stopping iteration based on the threshold value $\bar{\varepsilon}_{\nu|\mu,\psi}$ (among all values of $\mu \in \{0, 1, \ldots, \mu_{\max}\}$) for a given trial threshold value ψ. The green asterisk represents the position where the minimization argument in (5.3) is satisfied, so $\psi = \Upsilon$ for this position. Note that the value of Υ can be determined definitively, given this resulting illustration from simulation.

perhaps a higher threshold would be favorable over Υ, again the bias is towards more data, so this is decided to be acceptable. These results indicate that Υ should provide a reasonable threshold for the stopping criteria when using SDPA.

Figures 5.8 and 5.9 illustrate the use of permissive and restrictive thresholds, respectively, for SAGE. These figures are provided for comparison with Figure 5.7. Judging from inspection of the three figures, none of the three thresholds appear to have been determined all that well, though the results in Figure 5.6 would suggest otherwise for the optimal threshold. Again, it's possible that there just simply may not be enough instances where SAGE-based decomposition had valid results beyond the first few ensembles for decomposition. The optimal threshold probability of correct detection of 0.637 is 11.0% less than the permissive threshold probability of correct

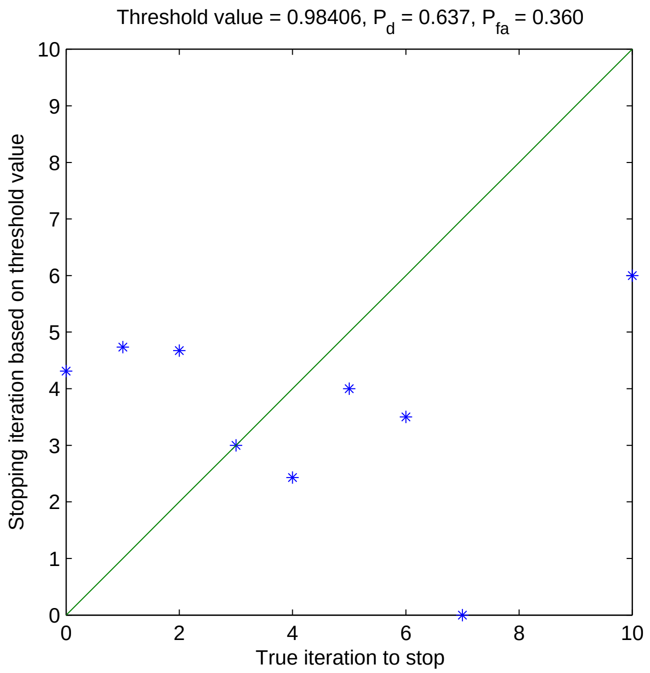

Figure 5.7: Illustration of the RMS error scenario by which the RMS error associated with the optimal threshold value $\Upsilon = 0.984063$ is obtained for SAGE. The blue asterisks are the mean stopping iteration values ($\bar{\varepsilon}_{\nu|\mu,\psi}$) given the true stopping iteration value μ. The green line illustrates the ideal relationship between the true iteration to stop and the mean stopping iteration based on the threshold value ψ. The closer the slope to unity of a line fit obtained from the values of $\bar{\varepsilon}_{\nu|\mu,\psi}$, the better the threshold, per (5.3). Note that the optimal threshold yields a probability of detection $P_d = 0.637$ and a probability of false alarm $P_{fa} = 0.360$.

detection in Figure 5.8, while the optimal threshold probability of false alarm is 13.3% less than the permissive threshold probability of false alarm. This indicates that the optimum threshold is indeed preferred over the permissive threshold. The optimal threshold probability of correct detection is 28.6% more than the restrictive threshold probability of correct detection in Figure 5.9, while the optimal threshold probability of false alarm is 18.8% less than the restrictive threshold probability of false alarm. Again, the optimum threshold is indeed the most preferable case for SAGE-based decomposition.

When comparing Figures 5.5 and 5.10 for SDPA and SAGE, respectively, the best ROC curve among several is generally considered to be the one for which inte-

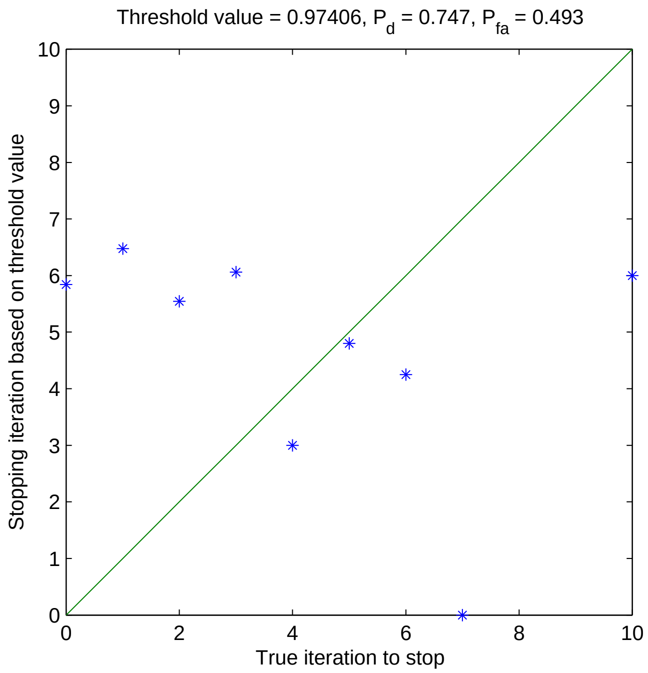

Figure 5.8: Illustration of the RMS error scenario by which the RMS error associated with a permissive threshold value of $\Upsilon - 0.01$ is obtained for SAGE. The blue asterisks are the mean stopping iteration values $(\bar{\varepsilon}_{\nu|\mu,\psi})$ given the true stopping iteration value μ. The green line illustrates the ideal relationship between the true iteration to stop and the mean stopping iteration based on the threshold value ψ. The closer the slope to unity of a line fit obtained from the values of $\bar{\varepsilon}_{\nu|\mu,\psi}$, the better the threshold, per (5.3). Note that this permissive threshold yields a probability of detection $P_d = 0.747$ and a probability of false alarm $P_{fa} = 0.493$.

gration under the ROC curve yields the highest number. This being the case, it is clear that the implementation of the stopping criteria is treated more favorably by SDPA, versus the SAGE-based algorithm, given the constraints placed on the two algorithms. The SDPA probability of correct detection for the optimum threshold of 0.812 is 17.5% higher than that for SAGE, while the probability of false alarm for SDPA of 0.401 is 4.1% higher. Furthermore, the weak ROC curve for SAGE raises the concern that it may not be appropriate to make use of a stopping criteria in the SAGE algorithm at all. In short, this data indicates that the stopping criteria is more suited for use in SDPA than SAGE. Further consideration of an appropriate stopping criteria for SAGE is recommended, but is not in the scope of this research.

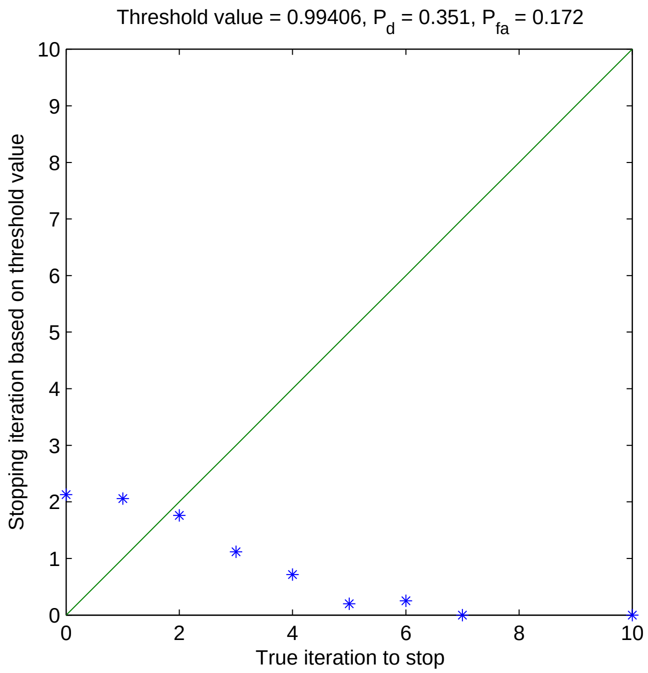

Figure 5.9: Illustration of the RMS error scenario by which the RMS error associated with a restrictive threshold value of $\Upsilon + 0.01$ is obtained for SAGE. The blue asterisks are the mean stopping iteration values $(\bar{\varepsilon}_{\nu|\mu,\psi})$ given the true stopping iteration value μ. The green line illustrates the ideal relationship between the true iteration to stop and the mean stopping iteration based on the threshold value ψ. The closer the slope to unity of a line fit obtained from the values of $\bar{\varepsilon}_{\nu|\mu,\psi}$, the better the threshold, per (5.3). Note that this restrictive threshold yields a probability of detection $P_d = 0.351$ and a probability of false alarm $P_{fa} = 0.172$.

Consideration of the comparative effectiveness of the stopping criteria in halting each of the two algorithms, SDPA and SAGE, would be a worthy objective. In the case of the results presented in this chapter, it is determined that a more appropriate stopping criteria should be established for SAGE before considering a head-to-head comparison of the algorithms with stopping criteria implemented.

5.6 Summary

This chapter outlined the methodology, the simulation scenario, and the results of the implementation of the stopping criteria in both SDPA and the SAGE-based algorithm. The results indicate that the stopping criteria did indeed perform as

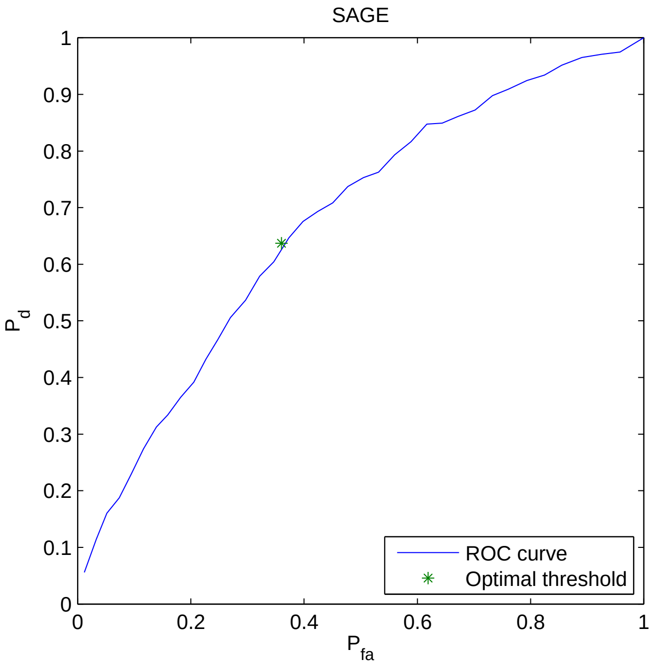

Figure 5.10: Receiver Operating Characteristic (ROC) curve computed using (5.6) and (5.8) for decomposition trials using SAGE. The blue curve is the ROC curve itself. The green asterisk indicates the position on the ROC curve associated with the optimal threshold value $\Upsilon = 0.984063$. The optimal threshold yields a probability of detection $P_d = 0.637$ and a probability of false alarm $P_{fa} = 0.360$. The ROC curve computation makes use of $(\varepsilon_p)_{final}$, which is bound by the the assumptions discussed in Section 4.2.4. As $(\varepsilon_p)_{final}$ increases, P_{fa} is reduced, so constraining $(\varepsilon_p)_{final}$ has the effect of distorting the true ROC curve, as computed using (5.6) and (5.8). Because of this distortion, the ROC curve illustrated in this figure should not be treated as conclusive. The purpose of including the ROC curve is to provide a comparative instrument between this figure and Figure 5.5.

desired in determining a suitable stopping criteria. In the case of the SAGE-based algorithm, further research is required to determine the suitability of the methodology outlined in this chapter within the algorithm.

VI. Conclusion

6.1 Summary

The research outlined in this dissertation involved the development of an algorithm, SDPA, which is used to estimate received multipath waveforms in GNSS signals. SDPA is described canonically in Chapter III. The purpose of this algorithm is to obtain multipath parameters which can be used to construct models for describing the effects that the local environment surrounding the GNSS receiver may have on a received GNSS signal. With these models, simulators could be constructed in order to generate data used to test receiver signal processing techniques.

The development of SDPA is a component of the two principle contributions to the community made with this research. The first is the development of SDPA itself and comparison with SAGE, which is believed to be the state-of-the-art in GNSS multipath parameterization, given the constraints on the problem. SDPA is shown through simulation to outperform the state-of-the-art in estimating the signal content within the search spaces that are associated with simulated generations of received GNSS data in some limited scenarios. More specifically, SDPA is shown in Chapter IV to outperform SAGE-based decomposition when simulated data of duration equal to the integration period of 1 msec and a value of C/N_0 less than or equal to 40 dB-Hz is considered. This finding exposes the existence of scenarios where decomposition of GNSS received signals is more appropriately performed using SDPA rather than with SAGE.

The second contribution to the community, the development of a stopping criteria to be used in the SDPA architecture and a comparison between SDPA and SAGE of the results of optimizing the threshold associated with the stopping criteria, is presented in Chapter V. This stopping criteria is used to halt a decomposition engine (be it SDPA or SAGE) at the point when parameterization of salient components within a recorded signal has been completed.

The SDPA requires further research in order to determine whether there are features of the algorithm that could be enhanced or applied differently from how it is used for the research presented in this dissertation. This research is presented in Section 6.2. Generally speaking, SDPA requires further development in order to effectively parameterize recorded signals in a wide variety of settings or circumstances.

6.2 Recommendations for Future Research

There are several recommendations for future research. Each of these recommendations is asserted with the objective of furthering the state of the art in the decomposition of recorded GNSS signals:

- The explicit use of SDPA in decomposing signals from GLONASS, Galileo, or Compass constellations. Section 2.10 presents discussion of the similarities and differences between the signals from each of the four GNSS constellations. With the exception of the use of frequency division multiplexing in GLONASS, the signal structures are very similar, and should only require variation in the matrix-vector multiplication operations used to decode received signal vectors in the eventual generation of search space matrices. Once the frequency division multiplexing in GLONASS is addressed through the use of multi-carrier receiver processing (which can be performed in software depending on the receiver hardware used in recording signals), it is believed that the variation in SDPA required to process signals from GNSS constellations other than GPS is minimal.
- The use of SDPA in an automated algorithm that makes use of multi-target tracking theory to track multipath ray waveform parameters over time. Findings of the initial research into the feasibility of SDPA in an automated tracking algorithm is discussed in [149].
- The explicit consideration of SDPA in the mitigation of multipath, as is done using MEDLL.
- The explicit consideration of SDPA in the estimation of parameters defining individual direct path and multipath ray waveforms, rather than estimating ray

waveform parameters with the single intent of replicating received search space matrices.

- The examination and implementation of emerging decomposition techniques, such as Empirical Mode Decomposition, in an algorithmic framework designed to decompose recorded GNSS signals. An example framework for signal decomposition is presented in Figure 3.2. This framework may require adaptation in order to accommodate the implementation of a decomposition technique differing from that developed for SDPA or for SAGE-based processing.
- The consideration of compressive sensing techniques to simplify the signals being decomposed, thus reducing the latency associated with decomposition and parameterization.
- The examination and integration of one-dimensional optimization methods making use of maximum likelihood estimation, such as that used in SAGE, in order to improve SDPA decomposition performance. Though it is shown in this dissertation that SDPA actually outperforms SAGE in certain scenarios, SAGE generally outperforms SDPA. SAGE has the advantage of using correlation results directly, in order to obtain relative propagation delay offset estimates. It is possible, as indicated by offline simulation, that SAGE may have an inherent advantage in the estimation of the relative propagation delay offset.
- The consideration of SDPA in the presence of interference that is not structured as a GNSS signal. Multipath is structured as a replica of a GNSS signal, but the presence of other types of interference in the environment is not considered in this research.
- The integration of GNSS receiver outputs in order to inform the SDPA on the structure of the signals being decomposed and parameterized.
- The integration of vision-based situational indicators into the SDPA. Use of these indicators may provide insight into the multipath scenario that may be

presented to the GNSS receiver, informing the SDPA design in the rapid estimation of multipath ray waveform parameters.

- The consideration of applicability to weighted search space content. Portions of the search space where signal content is high could be weighted in decomposition more heavily than areas where signal content is less. This may have the effect of improving search space estimates, particularly as the algorithm iterates.
- Further consideration of an appropriate stopping criteria for the SAGE-based algorithm. The stopping criteria outlined in Chapter V does not appear to perform as desired when using SAGE. Further research is required to determine why this is the case.
- The application of SDPA-based decomposition to generalized wireless communications signaling, particularly signals that make use of direct sequence spread spectrum waveform designs. There is a possibility that applying SDPA in a wireless communications receiver may yield improved multipath mitigation, relative to conventional multipath mitigation techniques employed in spread spectrum signaling.
- The consideration of SDPA-based decomposition in radar applications. Radar contends with the impact of multipath in radar receiver signal processing. It's possible that the use of a decomposition engine such as SDPA may improve multipath estimation and mitigation performance in sensing through the use of radar.

Bibliography

1. P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Lincoln, MA: Ganga-Jamuna Press, 2006.

2. J. G. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

3. B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

4. *Global Positioning System Standard Positioning Service Performance Standard*, 4th ed., ASD(NII)/DASD (C3, Space, and Spectrum), 2008.

5. *Global Positioning System Precise Positioning Service Performance Standard*, ASD(NII)/DASD (C3, Space, and Spectrum), 2007.

6. S. Gunawardena, "Development of a Transform-Domain Instrumentation Global Positioning System Receiver for Signal Quality and Anomalous Event Monitoring," Ph.D. dissertation, Dept. Elect. Eng., Ohio Univ., Athens, OH, 2007.

7. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

8. R. E. Thomas and A. J. Rosa, *The Analysis and Design of Linear Circuits*, 3rd ed. New York, NY: John Wiley & Sons, 2001.

9. M. S. Braasch, "Multipath Effects," in *Global Positioning System: Theory and Applications*, vol. 1, ed. B. Parkinson, J. Spilker, P. Alexrad, and P. Enge. Washington, DC: AIAA, 1996.

10. B. M. Hannah, "Modelling and Simulation of Multipath Propagation," Ph.D. dissertation, Queensland Univ. of Technology, Brisbane, Australia, 2001.

11. J. Weiss, S. Anderson, C. Fenwick, L. Stowe, and P. Axelrad, "Aircraft Carrier Multipath Modeling for Sea-Based JPALS," in *Proc. 18th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2005), Long Beach, CA, 2005, pp. 2697-2706.

12. K. Fan and X. Ding, "Estimation of GPS Carrier Phase Multipath Signals Based on Site Environment," *J. Global Positioning Systems*, vol. 5, no. 1, pp. 22-28, 2006.

13. R. Ercek, P. De Doncker, and F. Grenez, "Statistical Determination of the PR Error Due to NLOS-Multipath in Urban Canyons," in *Proc. 19th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2006), Fort Worth, TX, 2006, pp. 1771-1777.

14. S. R. Saunders, *Antennas and Propagation for Wireless Communication Systems.* Chichester, UK: Wiley, 1999.

15. *Simulating Multipath*, Spirent Communications, Paignton, Devon, UK.

16. A. Jahn, H. Bischl, and G. Hein, "Channel Characterisation for Spread Spectrum Satellite Communications," in *Proc. IEEE 4th Int. Symposium on Spread Spectrum Techniques and Applications* (ISSSTA'96), Mainz, Germany, 1996.

17. C. J. Hegarty, M. Tran, and J. W. Betz, "Multipath Performance of the New GNSS Signals," in *Proc. 2004 National Tech. Meeting of The Institute of Navigation*, San Diego, CA, 2004, pp. 333-342.

18. M. Irsigler, J. Avila-Rodgriguez, and G. Hein, "Criteria for GNSS Multipath Performance Assessment," in *Proc. 18th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2005), Long Beach, CA, 2005, pp. 2166-2177.

19. P. W. Ward, J. W. Betz, and C. J. Hegarty, in *GPS: Principles and Applications*, 2nd ed., ed. E.D. Kaplan and C.J. Hegarty. Norwood, MA: Artech House, 2006.

20. "Spirent Launches GPS Record and Playback System," *Inside GNSS*, Jan. 6, 2011.

21. G. Mylne, "The Effectiveness of a Multiple-Frequency Global Navigation Satellite System (GNSS) on Multipath Mitigation," B.S. Research Project, Faculty of Engineering and Surveying, Univ. of Southern Queensland, Toowoomba, Queensland, Australia, 2007.

22. U. Engel, "Improving Position Accuracy by Combined Processing of Galileo and GPS Satellite Signals," in *Proc. 11th Int. Conf. on Information Fusion* (FUSION), Cologne, France, 2008, pp. 1-8.

23. D. Skournetou and E. S. Lohan, "Non-coherent multiple correlator delay structures and their tracking performance for Galileo signals, in *Proc. European Navigation Conf. - Global Navigation Satellite Systems Conf.* (ENC-GNSS), Geneva, Switzerland, 2007.

24. E. S. Lohan, A. Lakhzouri, and M. Renfors, "Feedforward Delay Estimators in Adverse Multipath Propagation for Galileo and Modernized GPS Signals," *EURASIP J. Applied Signal Processing*, vol. 2006, pp. 157-175, Jan., 2006.

25. O. Rabaste, F. Brahim, and T. Chonavel, "An MCMC Algorithm for BOC and AltBOC Signaling Acquisition in Multipath Environments," in *Proc. IEEE/ION PLANS 2008*, Monterey, CA, 2008, pp. 424-432.

26. P. Xia and M. Ghogho, "Evaluation of Multiple Effects Interference Cancellation in GNSS using Space-Time based Array Processing," *Int. J. Control, Automation, and Systems*, vol. 6, no. 6, pp. 884-893, Dec., 2008.

27. A. Brown and B. Mathews, "GPS Multipath Mitigation Using a Three Dimensional Phased Array," in *Proc. 18th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2005), Long Beach, CA, 2005, pp. 659-666.

28. G. Seco-Granados, "Antenna Arrays for Multipath and Interference Mitigation in GNSS Receivers," Ph.D. dissertation, Dept. Signal Theory and Communications, Universitat Politecnica de Catalunya (Barcelona Tech), Barcelona, Spain, 2000.

29. G. Seco-Granados, J. Fernandez-Rubio, and C. Fernandez-Prades, "ML Estimator and Hybrid Beamformer for Multipath and Interference Mitigation in GNSS Receivers," *IEEE Trans. Signal Processing*, vol. 53, no. 3, pp. 1194-1208, Mar., 2005.

30. D. Nahimana, E. Duflos, and J. Marais, "Reception State Estimation of GNSS Satellites in Urban Environment using Particle Filtering," in *Proc. 11th Int. Conf. on Information Fusion* (FUSION), Cologne, France, 2008, pp. 1-5.

31. G. Seco-Granados and J. Fernandez-Rubio, "Multipath and Interference Errors Reduction in GNSS by Joint Pseudorange Measurement and Array Beamforming," in *Proc. 1st European Symposium on Global Navigation Satellite Systems* (GNSS), Munich, Germany, 1997, pp. 8-16.

32. O. Le Marchand, P. Bonnifait, J. Ibaez-Guzmn, D. Betaille, and F. Peyret, "Characterization of GPS Multipath for Passenger Vehicles Across Urban Environments," in *Proc. European Navigation Conf. - Global Navigation Satellite Systems Conf.* (ENC-GNSS), Naples, Italy, 2009.

33. Y. Yang, R. Hatch, and R. Sharpe, "GPS Multipath Mitigation in Measurement Domain and Its Applications for High Accuracy Navigation," in *Proc. 17th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2004), Long Beach, CA, 2004, pp. 1124-1130.

34. J. Marais, E. Duflos, N. Viandier, D. Nahimana, and A. Rabaoui, "Advanced Signal Processing Techniques for Multipath Mitigation in Land Transportation Environment," in *Proc. 13th Intl. IEEE Annual Conf. on Intelligent Transport Systems,* Madeira Island, Portugal, 2010, pp. 1480-1485.

35. F. Caron, E. Duflos, M. Davy, P. Vanheeghe, "Particle Filtering for Multipath Effects Reduction in Land Vehicule Positioning," in *Proc. IMACS Multiconference on Computational Engineering in Systems Applications,* Beijing, China, 2006, pp. 37-44.

36. N. Viandier, A. Rabaoui, J. Marais, and E. Duflos, "Enhancement of Galileo and Multi-constellation Accuracy by Modeling Pseudorange Noises," in *Proc. 9th Intl. Conf. on Intelligent Transport Systems Telecommunications* (ITST), Lille, France, 2009, pp. 459-464.

37. K. Fan and X. Ding, "Quantitative Estimation and Modelling of GPS Carrier Phase Multipath Signals," in *Proc. 6th Geomatic Week*, Barcelona, Spain, 2005.

38. M. Shah and Y. Lai, "Performance of Integer Parameter Estimation Algorithm for GPS Signals in Noisy Environment," in *Proc. 17th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2004), Long Beach, CA, 2004, pp. 166-174.

39. L. Wanninger and M. May, "Carrier Phase Multipath Calibration of GPS Reference Stations," in *Proc. 13th Int. Tech. Meeting of the Satellite Division of the Institute of Navigation* (ION GPS 2000), Salt Lake City, UT, 2000.

40. R. Leandro, M. Santos, and R. Langley, "Analyzing GNSS Data in Precise Point Positioning Software," *J. GPS Solutions*, vol. 15 no. 1, Jan., 2011.

41. U. Vollath and K. Sauer, "FAMCAR Approach for Efficient Multi-Carrier Ambiguity Estimation," in *Proc. European Navigation Conf. - Global Navigation Satellite Systems Conf.* (ENC-GNSS), Rotterdam, Netherlands, 2004.

42. R. Phelts and P. Enge, "The Case For Narrowband Receivers," in *Proc. 2000 National Tech. Meeting of The Institute of Navigation*, Anaheim, CA, 2000, pp. 511-520.

43. G. Lachapelle, M. Petovello, and J. Ray, "GNSS Solutions: Atomic Clocks on Satellites and Mitigating Multipath," *Inside GNSS*, pp. 22-27, Sep. 2006.

44. T. Peres, "Multipath Mitigation Techniques Suitable For Low Cost GNSS Receivers," M.S. thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, Lisbon, Portugal, 2008.

45. J. Soubielle, I. Fijalkow, P. Duvaut, and A. Bibaut, "GPS Positioning in a Multipath Environment," *IEEE Trans. Signal Processing*, vol. 50, no. 1, pp. 141-150, 2002.

46. A. Van Dierendonck, P. Fenton, and T. Ford, "Theory and Performance of Narrow Correlator Spacing in a GPS Receiver, *Navigation: J. Institute of Navigation*, vol. 39, no. 3, pp. 265-283, Fall, 1992

47. J. Wu and A. Dempster, "Multipath mitigation performance comparison of strobe correlators in GNSS receivers," in *Proc. Int. Symposium on GPS/GNSS*, Tokyo, Japan, 2008.

48. P. Fenton and J. Jones, "The Theory and Performance of NovAtel Inc.'s Vision Correlator," in *Proc. 18th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2005), Long Beach, CA, 2005, pp. 2178-2186.

49. M. Sahmoudi and M. Amin, "Improved Maximum-Likelihood Time Delay Estimation for GPS Positioning in Multipath, Interference and Low SNR Environments," in *Proc. IEEE/ION PLANS 2006*, San Diego, CA, 2006, pp. 876-882.

50. F. Linde, "Narrow Bandwidth Multipath Detection and Mitigation," M.S. thesis, Dept. Computer Science and Electrical Engineering, Lulea University of Technology, Lulea, Sweden, 2007.

51. J. Parro-Jimenez, J. Lopez-Salcedo, and G. Seco-Granados, "Multipath Detection Metrics and Attenuation Analysis Using a GPS Snapshot Receiver in Harsh Environments," in *Proc. European Conf. on Antennas and Propagation* (EuCAP), Berlin, Germany, 2009.

52. R. Hosbas, A. Pirti, and H. Erkaya, "Multipath and Multipath Reduction in the Urban Environments (Especially for L1 Signal Processing)," *Geodetski list*, vol. 4, pp. 301-313, 2009.

53. J. Sleewaegen and F. Boon, "Mitigating Short-Delay Multipath: a Promising New Technique," in *Proc. 14th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GPS 2001), Salt Lake City, UT, 2001, pp. 204-213.

54. R. van Nee, J. Sievereld, P. Fenton, and B. Townsend, "The Multipath Estimating Delay Lock Loop: Approaching Theoretical Accuracy Limits," in *Proc. IEEE Position, Location and Navigation Symposium*, Las Vegas, NV, 1994.

55. M. Sanchez-Fernandez, M. Aguilera-Forero, and A. Garcia-Armada, "Performance Analysis and Parameter Optimization of DLL and MEDLL in Fading Multipath Environments for Next Generation Navigation Receivers," *IEEE Trans. Consumer Electronics*, vol. 53, no. 4, pp. 1302-1308, Nov., 2007.

56. L. Weill, "Multipath Mitigation: How Good Can It Get with New Signals?" *GPS World*, 1 June 2003.

57. F. Antreich, O. Esbri-Rodriguez, J. Nossek, and W. Utschick, "Estimation of Synchronization Parameters Using SAGE in a GNSS-Receiver," in *Proc. 18th Intl. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2005), Long Beach, CA, 2005, pp. 2124-2131.

58. K. Rouabah and D. Chikouche, "GPS/Galileo Multipath Detection and Mitigation Using Closed-Form Solutions," *Mathematical Problems in Engineering*, vol. 2009, 2009.

59. F. Lehmann. "Deterministic Particle Filtering for GPS Navigation in the Presence of Multipath," in *Proc. 15th European Signal Processing Conf.* (EUSIPCO), Poznan, Poland, 2007.

60. P. Closas, C. Fernandez-Prades, J. Fernandez-Rubio, A. Ramirez-Gonzalez, "Multipath Mitigation Using Particle Filtering," in *Proc. 19th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2006), Fort Worth, TX, 2006, pp. 1733-1740.

61. P. Closas, C. Fernandez-Prades, and J. Fernandez, "Sequential Monte-Carlo Approximation to the ML Time-Delay Estimator in a Multipath Channel," in *Proc. Int. Workshop on Signal Processing Advances in Wireless Communications* (SPAWC), Cannes, France, 2006.

62. X. Liu, P. Closas, J. Liu, and X. Hu, "Particle Filtering and its Application for Multipath Mitigation with GNSS Receivers," in *Proc. IEEE/ION PLANS 2010*, Indian Wells, CA, 2010, pp. 1168-1173.

63. C. Yang and A. Porter, "Frequency-Domain Characterization of GPS Multipath for Estimation and Mitigation," in *Proc. 18th Int. Tech. Meeting of the Satellite*

Division of The Institute of Navigation (ION GNSS 2005), Long Beach, CA, 2005, pp. 2104-2118.

64. D. Yang, Y. Yao, and J. Guo, "GPS Multipath Signal Estimation Based on All Phase DFT," *Chinese J. Electronics*, vol. 19, no. 2, Apr., 2010.

65. W. Nam and S. Kong, "Modified Least-Squares based Iterative Multipath Super-Resolution Algorithm," in *Proc. 2011 Int. Tech. Meeting of The Institute of Navigation*, San Diego, CA, 2011, pp. 591-595.

66. Z. Shengkang, W. Hongbo, Y. Jun and H. Leiming, "GPS Short-delay Multipath Estimation and Mitigation Based on Least Square Method," *J. Systems Engineering and Electronics*, vol. 20, no. 5, pp. 954-961, 2009.

67. K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach.* Boston, MA: Birkhauser, 2007.

68. A. Steingass and A. Lehner, "Measuring GALILEOs multipath channel," in *Proc. European Navigation Conf. - Global Navigation Satellite Systems Conf.* (ENC-GNSS), Graz, Austria, 2003.

69. A. Steingass and A. Lehner, "Measuring the Navigation Multipath Channel...A Statistical Analysis," in *Proc. 17th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2004), Long Beach, CA, 2004, pp. 1157-1164.

70. A. Lehner and A. Steingass, "A Novel Channel Model for Land Mobile Satellite Navigation," in *Proc. 18th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2005), Long Beach, CA, 2005, pp. 2132-2138.

71. F. Schubert, R. Prieto-Cerdeira, and A. Steingass, "GNSS Software Simulation System for Realistic High-Multipath Environments," in *Proc. 4th European Space Agency Workshop on Satellite Navigation User Equipment Technologies* (NAVITEC 2008), Noordwijk, The Netherlands, 2008.

72. A. Lehner, A. Steingass, and F. Schubert, "A location and movement dependent GNSS multipath error model for pedestrian applications," *ATTI J. dell'Istituto Italiano di Navigazione.* pp. 108-119, 2009.

73. D.N. Schettino, F.J.S. Moreira, and C.G. Rego, "Efficient Ray Tracing for Radio Channel Characterization of Urban Scenarios," *IEEE Trans. Magnetics*, vol. 43, no. 4, pp. 1305-1308, 2007.

74. F. Schubert and B. Krach, "Simulation of High-Realistic Multipath Environments: Developments and Applications," in *Proc. 2nd EuNavTec Positions Congress* (POSITIONs 2008), Dresden, Germany, 2008.

75. S. N. Sadrieh, A. Broumandan, and G. Lachapelle, "Spatial/Temporal Characterization of the GNSS Multipath Fading Channels," in *Proc. 23rd Int. Tech.*

Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010), Portland, OR, 2010, pp. 393-401.

76. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley & Sons, 2001.

77. J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* Hoboken, NJ: John Wiley & Sons, 2003.

78. S. C. Gustafson. EENG 620. Class Lecture, Topic: Simulated Annealing. Dept. Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Feb., 2006.

79. A. Goldsmith, *Wireless Communications.* Cambridge, UK: Cambridge Univ. Press, 2005.

80. W. Weibing, Z. Quiping, and X. Xinyi, "Rayleigh Channel Estimation Based on Particle Filter with Simulated Annealing Algorithm," *Wuhan Univ. J. Natural Sciences*, vol. 14, no. 4, pp. 153-157, 2009.

81. P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1. Arlington, VA: Navtech Book and Software Store, 1994.

82. Z. Zhi-Jin, Z. Shi-Lian, X. Chun-Yun, and K. Xian-Zheng, "Discrete channel modelling based on genetic algorithm and simulated annealing for training hidden Markov model," *Chinese Physics*, vol. 16, no. 6, pp. 1619-1623, Jun., 2007.

83. M. Gao, J. Tian, and Y. Sun, "Research of Wireless Fading Channel Modeling Based on Improved Simulated Annealing Neural Network," in *Proc. 2009 IEEE Int. Conf. on Mechatronics and Automation*, Changchun, China, 2009, pp. 2707-2711.

84. A. T. Zimmerman and J. P. Lynch, "Parallelized Simulated Annealing for Model Updating in Ad-Hoc Wireless Sensing Networks," in *Proc. Int. Workshop on Data Intensive Sensor Networks*, Mannheim, Germany, 2007.

85. A. P. Iserte, "Channel State Information and Joint Transmitter-Receiver Design in Multi-Antenna Systems," Ph.D. dissertation, Universitat Politecnica de Catalunya (Barcelona Tech), Barcelona, Spain, 2004.

86. S. Baselga and L. Garcia-Asenjo, "Global robust estimation and its application to GPS positioning," *Computers and Mathematics with Applications*, vol. 56, no. 3, pp. 709-714, 2008.

87. S. Baselga, "Global Optimization Applied to GPS Positioning by Ambiguity Functions," *Measurement Science and Technology*, vol. 21, no. 12, 2010.

88. M. Staudinger, "A Cost Orientated Approach to Geodetic Network Optimisation," Ph.D. dissertation, Faculty of Tech. Science, Univ. of Technology, Vienna, Vienna, Austria, 1999.

89. G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2008.

90. Y. Chen and M. Gupta, "EM Demystified: An Expectation-Maximization Tutorial," Dept. Elect. Eng., Univ. Washington, Seattle, WA, UWEE Tech. Rep. UWEETR-2010-0002, 2010.

91. M. Feder and E. Weinstein, "Optimal Multiple Source Location Estimation via the EM Algorithm," in *Proc. 1985 Int. Conf. Acoust., Speech, Signal Processing* (ICASSP 1985), Tampa Bay, FL, 1985, pp. 1762-1765.

92. M. Feder and E. Weinstein, "Multipath and Multiple Source Array Processing via the EM Algorithm," in *Proc. 1986 Int. Conf. Acoust., Speech, Signal Processing* (ICASSP 1986), Tokyo, Japan, 1986, pp. 2503-2506.

93. M. Feder and E. Weinstein, "Parameter Estimation of Superimposed Signals Using the EM Algorithm," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 36, no. 4, pp. 477-489, Apr., 1988.

94. J. Fessler and A. Hero, "Space-Alternating Generalized Expectation-Maximization Algorithm," *IEEE Trans. Signal Processing*, vol. 42, no. 10, pp. 2664-2677, Oct., 1994.

95. F. Dietrich, *Robust Signal Processing for Wireless Communications.* Berlin, Germany: Springer-Verlag, 2008.

96. B. Fleury, D. Dahlhaus, R. Heddergott, and M. Tschudin, "Wideband Angle of Arrival Estimation Using the SAGE Algorithm," in *Proc. IEEE 4th Int. Symp. on Spread Spectrum Techniques and Application* (ISSSTA 1996), Mainz, Germany, 1996, pp. 79-85.

97. K. Pedersen, B. Fleury, and P. Mogensen, "High Resolution of Electromagnetic Waves in Time-Varying Radio Channels," in *Proc. 8th IEEE Intl. Symp. on Personal, Indoor and Mobile Radio Communications* (PIMRC 1997), 1997, pp. 650-654.

98. B. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. Pedersen, "Channel Parameter Estimation in Mobile Radio Environments Using the SAGE Algorithm," *IEEE J. Selected Areas in Communications*, vol. 17, no. 3, pp. 434-450, Mar., 1999.

99. J. Verhaevert, E. Van Lil, and A. Van de Capelle, "Direction of Arrival (DOA) Parameter Estimation with the SAGE Algorithm," *Signal Processing*, vol. 84, no. 3, Apr., 2003.

100. P. Stoica and R. Moses, *Spectral Analysis of Signals.* Upper Saddle River, NJ: Prentice Hall, 2005.

101. I. Groh, S. Sand, and C. Mensing, "Complexity Reduced Multipath Mitigation in GNSS with the GRANADA Bit-True Software Receiver," in *Proc. IEEE/ION PLANS 2008*, Monterey, CA, 2008, pp. 418-423.

102. X. Hu, E. S. Lohan, I. Groh, S. Sand, and M. Renfors, "Multiple Branch Delay Lock Loop Comparison for SinBOC(1,1) Signal Tracking in Multipath Environments," in *Proc. 2009 Int. Tech. Meeting of The Institute of Navigation*, Anaheim, CA, 2009, pp. 434-441.

103. O. Esbri-Rodriguez, M. Philippakis, A. Konovaltsev, F. Antreich, C. Martel, and D. Moore, "Antenna-based Multipath and Interference Mitigation for Aeronautical Applications: Present and Future," in *Proc. 19th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2006), Fort Worth, TX, 2006, pp. 754-762.

104. S. Rougerie, G. Carrie, L. Ries, F. Vincent, R. Pascaud, E. Corbel, and M. Monnerat, "Multipath Mitigation Methods Based on Antenna Array," in *Proc. 2011 Int. Tech. Meeting of The Institute of Navigation*, San Diego, CA, 2011, pp. 596-605.

105. S. Rougerie, G. Carrie, F. Vincent, L. Ries, and M. Monnerat, "A New Multipath Mitigation Method for GNSS Receivers Based on an Antenna Array," *Intl. J. Navigation and Observation*, vol. 2012, Mar., 2012.

106. L. R. Weill, "GNSS Solutions: Difference between Signal Acquisition and Tracking," *Inside GNSS*, pp. 22-27, Jan./Feb. 2011.

107. M. S. Braasch, "GNSS Solutions: Signal Acquisition and Search, and Antenna Polarization," *Inside GNSS*, pp. 26-32, Mar./Apr. 2007.

108. S. Semmelrodt, "Spectral Analysis and Linear Prediction Toolbox for Stationary and Non-Stationary Signals," *Frequenz*, vol. 58, issue 7-8, pp. 185-187, Jul., 2004.

109. *fminbnd* [Online]. Available:
`http://www.mathworks.com/help/techdoc/ref/fminbnd.html`.

110. M. Sahmoudi and R. Landry, "Multipath Mitigation Techniques Using Maximum-Likelihood Principle," *Inside GNSS*, Nov/Dec 2008.

111. H. Van Trees, *Detection, Estimation, and Modulation Theory*, part 1. New York, NY: John Wiley & Sons, 2001.

112. H.V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York, NY: Springer, 1994.

113. D. McNicol, *A Primer of Signal Detection Theory.* Mahwah, NJ: Lawrence Erlbaum Associates, 2005.

114. R. van Nee, "The Multipath Estimating Delay Lock Loop," in *Proc. IEEE 2nd Intl. Symp. Spread Spectrum Techniques and Applications* (ISSSTA), Yokohama, Japan, 1992.

115. A. Hewitt, W. Lau, J. Austin, and E. Vilar, "An Autoregressive Approach to the Identification of Multipath Ray Parameters from Field Measurements," *IEEE Trans. Signal Processing*, vol. 37, no. 11, pp. 1136-1143, Nov., 1989.

116. A. van der Veen, M. Vanderveen, and A. Paulraj, "Joint Angle and Delay Estimation Using Shift-Invariance Techniques," *IEEE Trans. Signal Processing*, vol. 46, no. 2, pp. 405-418, Feb., 1998.

117. T. Lo, J. Litva, and H. Leung, "A New Approach for Estimating Indoor Radio Propagation Characteristics," *IEEE Trans. Antennas and Propagation*, vol. 42, no. 10, pp. 1369-1376, Oct., 1994.

118. M. Vanderveen, A. van der Veen, and A. Paulraj, "Estimation of Multipath Parameters in Wireless Communications," *IEEE Trans. Signal Processing*, vol. 46, no. 3, pp. 682-690, Mar., 1998.

119. B. Jeffs, "MIMO Wireless Multipath Ray Parameter Estimation from Channel Transfer Matrix Measurements," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing* (ICASSP), Orlando, FL, 2002, pp. 2345-2348.

120. J. Vera, "Efficient Multipath Mitigation in Navigation Systems," Ph.D. dissertation, Dept. Signal Theory and Communications, Univ. Politecnica de Catalunya, Barcelona, Spain, 2003.

121. J. Landel, "End-Effect, Stopping Criterion, Mode Mixing and Confidence Limit for the Hilbert-Huang Transform," M.S. thesis, Dept. Mech. Eng., National Univ. Singapore, Singapore, 2008.

122. Q. Xie, B. Xuan, S. Peng, J. Li, W. Xu, and H. Han, "Bandwidth Empirical Mode Decomposition and Its Application," *Intl. J. Wavelets, Multiresolution and Information Processing*, vol. 6, no. 6, pp. 777-798, 2008.

123. Q. Wu and S. Riemenschneider, "Boundary Extension and Stop Criteria for Empirical Mode Decomposition," *Advances in Adaptive Data Analysis*, vol. 2, no. 2, pp. 157-169, 2010.

124. K. Qi, Z. He, and Y. Zi, "Cosine Window-based Boundary Processing Method for EMD and Its Applications in Rubbing Fault Diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, pp. 2750-2760, May, 2007.

125. K. Shafqat, S. Pal, S. Kumari, and P. Kyriacou, "Empirical Mode Decomposition (EMD) Analysis of HRV Data from Locally Anesthetized Patients," in *Proc. 31st Ann. Intl. Conf. of IEEE EMBS*, Minneapolis, MN, 2009, pp. 2244-2247.

126. Y. Lu, A. Kasaeifard, E. Oruklu, and J. Saniie, "Fractional Fourier Transform for Ultrasonic Chirplet Signal Decomposition," *Advances in Acoustics and Vibration*, vol. 2012, May, 2012.

127. H. Najaf-Zadeh, R. Pichevar, H. Lahdili, and L. Thibault, "Perceptual Matching Pursuit for Audio Coding," in *Proc. 124th Conv. Audio Engineering Society*, Amsterdam, The Netherlands, 2008.

128. C. Coelho, S. Das, and A. Chattopadhyay, "Binary Tree SVM Based Framework for Mining Fatigue Induced Damage Attributes in Complex Lug Joints," *Proc.*

SPIE Modeling, Signal Processing, and Control for Smart Structures, vol. 6926, 2008.

129. L. Lovisolo, E. da Silva, M. Rodrigues, and P. Diniz, "Efficient Coherent Adaptive Representations of Monitored Electric Signals in Power Systems Using Damped Sinusoids," *IEEE Trans. Signal Processing*, vol. 53, no. 10, pp. 3831-3846, Oct., 2005.

130. J. Mairal, G. Sapiro, and M. Elad, "Learning Multiscale Sparse Representations for Image and Video Restoration," *Multiscale Model. Simul.*, vol. 7, no. 1, pp. 214-241, 2008.

131. S. Thombre, E. S. Lohan, H. Hurskainen, J. F. Raquet, and J. Nurmi, "Software-Based GNSS Signal Simulators - Past, Present, and Possible Future," in *Proc. European Navigation Conf. on Global Navigation Satellite Systems*, Braunschweig, Germany, 2010.

132. A. Nunes, T. Ferreira, J. Borrs, F. Nunes, F. Sousa, and G. Seco, "Signal Generator and Receiver Toolbox for Galileo/GPS Signals," in *Proc. 2nd ESA Workshop on Satellite Navigation*, Noordwijk, The Netherlands, 2004.

133. O. Julien, B. Zheng, L. Dong, and G. Lachapelle, "A Complete Software-Based IF GNSS Signal Generator for Software Receiver Development," in *Proc. 17th Int. Tech. Meeting of the Satellite Division of The Institute of Navigation* (ION GNSS 2004), Long Beach, CA, 2004, pp. 2146-2157.

134. C. Seynat, A. Kealy, and K. Zhang, "A Performance Analysis of Future Global Navigation Satellite Systems," *J. Global Positioning Systems*, vol. 3, no. 1-2, pp. 232-241, 2004.

135. C. Ouzeau and J. Korsakissok, "Performance Assessment of a Juzzle-Based GNSS Simulator," in *Proc. ENC GNSS 2009*, Naples, Italy, 2009.

136. A. M. Smith, "Global Navigation Satellite System Signal Simulator - An Analysis of the Effects of the Local Environment and Atmosphere on Receiver Positioning," Ph.D. dissertation, Univ. of Bath, Bath, UK, 2007.

137. *BeiDou Navigation Satellite System Signal In Space Interface Control Document (Test Version)*, China Satellite Navigation Office, 2011.

138. *European GNSS (Galileo) Open Service Signal In Space Interface Control Document,* European Union, 2010.

139. *GLONASS Interface Control Document*, Version 5.1. Moscow, Russia, 2002.

140. J. McGinthy, "Global Navigation Satellite System Software Defined Radio," M.S. thesis, Dept. Elect. Eng., Air Force Inst. Tech., Wright-Patterson AFB, OH, 2010.

141. M. Haker and J. Raquet, "Estimating Multipath in GNSS Signals Through a Novel Stochastic Search and Decomposition Algorithm," in *Proc. 24th Int. Tech.*

Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, 2011, pp. 1162-1172.

142. M. Haker and J. Raquet, "Applying Detection Theory to Define Stopping Criteria for the Signal Decomposition and Parameterization Algorithm," in *Proc. Intl. Conf. Localization and GNSS* (ICL-GNSS), Starnberg, Germany, 2012.

143. D. Borio, "A Statistical Theory for GNSS Signal Acquisition," Ph.D. dissertation, Politecnico di Torino, Torino, Italy, 2008.

144. J. Tsui, *Fundamentals of Global Positioning System Receivers: A Software Approach.* New York, NY: Wiley-Interscience, 2004.

145. A.J. Van Dierendonck, "GPS Receivers," in *Global Positioning System: Theory and Applications*, vol. 1, ed. B. Parkinson, J. Spilker, P. Alexrad, and P. Enge. Washington, DC: AIAA, 1996.

146. A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. Reading, MA: Addison-Wesley, 1993.

147. D. J. Moelker, "Interference in Satellite Navigation and Mobile Communication," Ph.D. dissertation, Delft Univ. Tech., Delft, Netherlands, 1998.

148. *DLR SatNav Model Page.* [Online]. Available:
`http://www.kn-s.dlr.de/satnav/LandMobile.html`.

149. M. Haker and J. Raquet, "Tracking Multipath in Received GNSS Signals through use of a Signal Decomposition and Parameterization Algorithm," in *Proc. 25th Int. Tech. Meeting of The Satellite Division of the Institute of Navigation* (ION GNSS 2012), Nashville, TN, 2012.

Vita

Major Marshall E. Haker graduated from Washington High School in Vinton, Iowa in 1993. He subsequently enlisted in the United States Air Force and served at the National Air and Space Intelligence Center at Wright-Patterson AFB, Ohio. In 1999, he was selected for the Airman Education and Commissioning Program. In 2002, he graduated with honors with a Bachelor of Science in Electrical and Computer Engineering from The Ohio State University, as well as the United States Air Force Officer Training School (OTS). Upon graduation from OTS, he was assigned to the Oklahoma City Air Logistics Center. He first served there as an avionics systems engineer in what was then named the C/KC-135 System Program Office. In 2004, he was assigned to serve as a depot maintenance engineer in the 76th Commodities Maintenance Group. In this assignment, Marshall led the modernization of the electronic signals processing hardware onboard radome test stands used to verify correct operation of B-52, KC-135, and E-3 radomes. The performance of the team responsible for this modernization led to their being awarded the 2006 Chief of Staff Team Excellence Award. Marshall was subsequently selected to pursue a Masters of Science in Electrical Engineering at the Air Force Institute of Technology (AFIT) in preparation for an assignment in the Air Force Research Laboratory (AFRL) Sensors Directorate. At AFRL, he served as an electronic warfare techniques development engineer. He was then selected to lead flight test for the second spiral of development of an airborne persistent surveillance platform that was at the time operating overseas in support of ongoing military operations in the Middle East. The performance of the team responsible for the development and test of the spiral upgrade led to their being awarded the 2009 AFRL Commander's Cup. In August 2009 he entered the doctoral program in the Graduate School of Engineering and Management at AFIT. Upon graduation he will serve as an assistant professor at AFIT.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
18 Dec 2012

2. REPORT TYPE
Dissertation

3. DATES COVERED *(From - To)*
10 Aug 2009 - 18 Dec 2012

4. TITLE AND SUBTITLE
Modeling the Effects of the Local Environment on a Received GNSS Signal

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Haker, Marshall E, Maj

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/DEE/ENG/12-11

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory (AFRL)/RYMN
Dr. Eric T. Vinande
2241 Avionics Circle
Wright-Patterson AFB OH 45433
(937) 938-4400 eric.vinande@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYMN

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

There is an ongoing need in the GNSS community for the development of high-fidelity simulators which generate data that replicates what can truly be expected from a challenging environment such as an urban canyon or an indoor environment. The algorithm developed for use in the research in this dissertation, the Signal Decomposition and Parameterization Algorithm (SDPA), is presented in order to respond to this need. This algorithm is designed to decompose a signal received using a GNSS recording and playback system and output parameters that can be used to reconstruct the effects on the signal of the environment local to the receiver at the time of recording. The SDPA itself is presented and compared with what is believed to be the state-of-the-art in GNSS multipath parameterization, a Space Alternating Generalized Expectation Maximization (SAGE) algorithm. The development and characterization of a stopping criteria that can be used to halt the SDPA when parameterization of salient components within a recorded signal has been completed.

15. SUBJECT TERMS
Global Positioning System, Global Navigation Satellite System, Multipath, Waveform, Estimation, Simulated Annealing, Expectation Maximization, Stopping criteria

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. John F. Raquet
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	177	(937) 255-3636 x4580 john.raquet@afit.edu